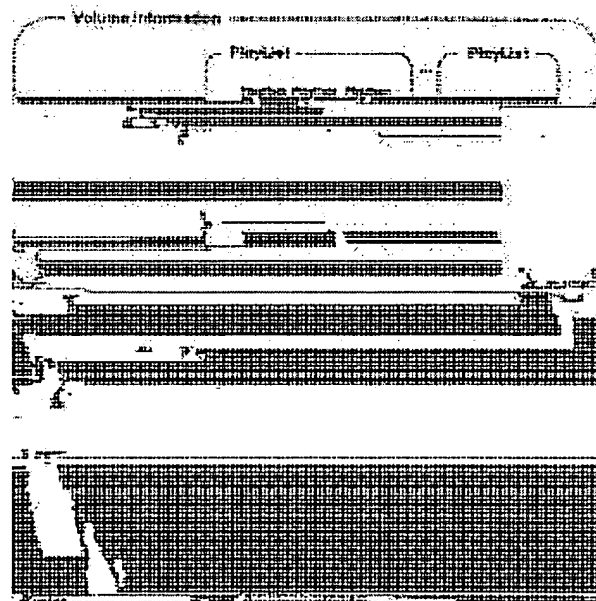


**INFORMATION PROCESSOR AND PROCESSING METHOD, AND
RECORDING MEDIUM THEREFOR, AND PROGRAM AND ITS RECORDING
MEDIUM**

Publication number: JP2002158971 (A)
Publication date: 2002-05-31
Inventor(s): KATO MOTOKI; HAMADA TOSHIYA
Applicant(s): SONY CORP
Classification:
- international: **H04N5/92; G11B20/12; G11B27/00; H04N5/91; H04N5/92; G11B20/12; G11B27/00; H04N5/91; (IPC1-7): H04N5/92; G11B20/12; G11B27/00; H04N5/91**
- European:
Application number: JP20010089959 20010327
Priority number(s): JP20010089959 20010327; JP20000183770 20000421; JP20000268042 20000905

Abstract of JP 2002158971 (A)

PROBLEM TO BE SOLVED: To retrieve a specified mark quickly. **SOLUTION:** An AV stream, as an entity of contents, is managed by ClipInformation and reproduction of the AV stream is managed by PlayList. As the attribute information of the AV stream, address information on discontinuity points in the AV stream RSPN arrival time discontinuity, information on associating the time information and address information in the AV stream EP map, TU map, and time information on a characteristic image in the AV stream ClipMark are recorded in the ClipInformation.



Data supplied from the esp@cenet database — Worldwide

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-158971

(P2002-158971A)

(43) 公開日 平成14年5月31日 (2002. 5. 31)

(51) Int.Cl. ⁷	識別記号	F I	テ-マコード*(参考)
H 0 4 N 5/92		G 1 1 B 20/12	5 C 0 5 3
G 1 1 B 20/12			1 0 3 5 D 0 4 4
	1 0 3	27/00	D 5 D 1 1 0
27/00		H 0 4 N 5/92	H
H 0 4 N 5/91		5/91	N
審査請求 未請求 請求項の数19 O L (全 75 頁)			

(21) 出願番号 特願2001-89959(P2001-89959)
(22) 出願日 平成13年3月27日(2001. 3. 27)
(31) 優先権主張番号 特願2000-183770(P2000-183770)
(32) 優先日 平成12年4月21日(2000. 4. 21)
(33) 優先権主張国 日本(J P)
(31) 優先権主張番号 特願2000-268042(P2000-268042)
(32) 優先日 平成12年9月5日(2000. 9. 5)
(33) 優先権主張国 日本(J P)

(71) 出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号
(72) 発明者 加藤 元樹
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(72) 発明者 浜田 俊也
東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内
(74) 代理人 100082131
弁理士 稲本 義雄

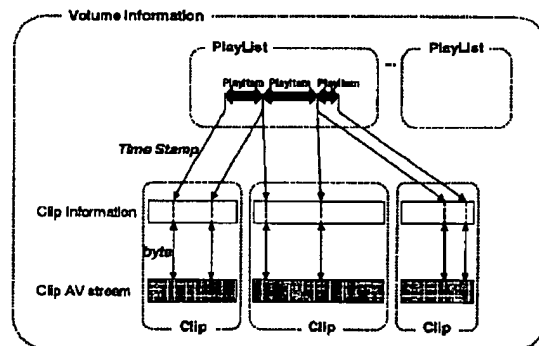
最終頁に続く

(54) 【発明の名称】 情報処理装置および方法、記録媒体、プログラム、並びに記録媒体

(57) 【要約】

【課題】 所定のマークを迅速に検索できるようにする。

【解決手段】 コンテンツの実体としてのAVストリームは、ClipInformationにより管理され、AVストリームの再生は、PlayListにより管理される。AVストリームの属性情報としての、AVストリーム中の不連続点のアドレス情報RSPN_arrival_time_discontinuity、AVストリーム中の時刻情報とアドレス情報を関連づける情報EP_map、TU_map、並びに、AVストリーム中の特徴的な画像の時刻情報ClipMarkは、ClipInformationに記録される。



【特許請求の範囲】

【請求項1】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置において、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成手段と、前記生成手段により生成された情報を前記Clip情報として前記記録媒体に記録する記録手段とを備えることを特徴とする情報処理装置。

【請求項2】 前記符号化情報が連続な区間の開始アドレス情報は、STCシーケンスまたはプログラムシーケンスの開始アドレスであり、前記時刻情報とアドレス情報を関連づける情報は、EP_mapまたはTU_mapであり、前記特徴的な画像の時刻情報は、ClipMarkであることを特徴とする請求項1に記載の情報処理装置。

【請求項3】 前記記録手段は、前記AVストリームの記録レートの平均値に関する情報を前記記録媒体にさらに記録することを特徴とする請求項1に記載の情報処理装置。

【請求項4】 前記平均値に関する情報は、TS_average_rateであることを特徴とする請求項3に記載の情報処理装置。

【請求項5】 前記AVストリームはトランスポートストリームであることを特徴とする請求項1に記載の情報処理装置。

【請求項6】 前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、前記トランスポートストリームのシステムタイムクロックが連続な区間であるSTCシーケンスの開始アドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項7】 前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、前記トランスポートストリームのプログラム内容が一定な区間であるプログラムシーケンスの開始アドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項8】 前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、前記トランスポートストリームのトランスポートパケットの到着時間に基づくアライバルタイムが連続な区間の開始アドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項9】 前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、前記トランスポートストリームのIピクチャのアドレスとそのプレゼンテーションタイムスタンプを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項10】 前記AVストリーム中の時刻情報とアドレス情報を関連づける情報の少なくとも一部を圧縮する圧縮手段をさらに備え、

前記記録手段は、前記圧縮手段により圧縮された前記情報を記録することを特徴とする請求項9に記載の情報処理装置。

【請求項11】 前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、トランスポートパケットの到着時刻に基づいたアライバルタイムと、それに対応するトランスポートパケットの前記AVストリームデータ中のアドレスを含むことを特徴とする請求項1に記載の情報処理装置。

【請求項12】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置の情報処理方法において、

前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、前記生成ステップにより生成された情報を前記Clip情報として前記記録媒体に記録する記録ステップとを含むことを特徴とする情報処理方法。

【請求項13】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置のプログラムにおいて、

前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、前記生成ステップにより生成された情報を前記Clip情報として前記記録媒体に記録する記録ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項14】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録される記録媒体に対して前記AVストリームを記録する情報処理装置を制御するコンピュータに、

前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、前記生成ステップにより生成された情報を前記Clip情報

として前記記録媒体に記録する記録ステップとを実行させるプログラム。

【請求項15】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置において、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生手段と、前記再生手段により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御手段とを備えることを特徴とする情報処理装置。

【請求項16】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置の情報処理方法において、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、前記再生ステップの処理により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御ステップとを含むことを特徴とする情報処理方法。

【請求項17】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置のプログラムにおいて、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、前記再生ステップの処理により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御ステップとを含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項18】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体から前記AVストリームを再生する情報処理装置を制御するコンピュータに、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始ア

ドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、前記再生ステップの処理により再生された前記Clip情報に基づいて、前記AVストリームの出力を制御する制御ステップとを実行させるプログラム。

【請求項19】 AVストリーム、前記AVストリームの実体を管理するClip情報、および前記AVストリームの再生を管理するPlayList情報が記録されている記録媒体において、前記Clip情報として、前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報、前記AVストリーム中の時刻情報とアドレス情報を関連づける情報、および前記AVストリーム中の特徴的な画像の時刻情報が記録されていることを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は情報処理装置および方法、記録媒体、並びにプログラムに関し、特に、AVストリーム内のIピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録する情報処理装置および方法、記録媒体、並びにプログラムに関する。

【0002】

【従来の技術】近年、記録再生装置から取り外し可能なディスク型の記録媒体として、各種の光ディスクが提案されつつある。このような記録可能な光ディスクは、数ギガバイトの大容量メディアとして提案されており、ビデオ信号等のAV(Audio Visual)信号を記録するメディアとしての期待が高い。この記録可能な光ディスクに記録するデジタルのAV信号のソース(供給源)としては、CSデジタル衛星放送やBSデジタル放送があり、また、将来はデジタル方式の地上波テレビジョン放送等も提案されている。

【0003】ここで、これらのソースから供給されるデジタルビデオ信号は、通常MPEG(Moving Picture Experts Group)2方式で画像圧縮されているのが一般的である。また、記録装置には、その装置固有の記録レートが定められている。従来の民生用映像蓄積メディアで、デジタル放送のデジタルビデオ信号を記録する場合、アナログ記録方式であれば、デジタルビデオ信号をデコード後、帯域制限をして記録する。あるいは、MPEG1 Video、MPEG2 Video、DV方式をはじめとするデジタル記録方式であれば、1度デコードされた後に、その装置固有の記録レート・符号化方式で再エンコードされて記録される。

【0004】しかしながら、このような記録方法は、供給されたビットストリームを1度デコードし、その後で帯域制限や再エンコードを行って記録するため、画質の

劣化を伴う。画像圧縮されたデジタル信号の記録をする場合、入力されたデジタル信号の伝送レートが記録再生装置の記録レートを超えない場合には、供給されたビットストリームをデコードや再エンコードすることなく、そのまま記録する方法が最も画質の劣化が少ない。ただし、画像圧縮されたデジタル信号の伝送レートが記録媒体としてのディスクの記録レートを超える場合には、記録再生装置でデコード後、伝送レートがディスクの記録レートの上限以下になるように、再エンコードをして記録する必要がある。

【0005】また、入力デジタル信号のビットレートが時間により増減する可変レート方式によって伝送されている場合には、回転ヘッドが固定回転数であるために記録レートが固定レートになるテープ記録方式に比べ、1度バッファにデータを蓄積し、バースト的に記録ができるディスク記録装置の方が記録媒体の容量をより無駄なく利用できる。

【0006】以上のように、デジタル放送が主流となる将来においては、データストリーマのように放送信号をデジタル信号のまま、デコードや再エンコードすることなく記録し、記録媒体としてディスクを使用した記録再生装置が求められると予測される。

【0007】

【発明が解決しようとする課題】上述したような装置により、複数のデータ（例えば、映像データや音声データなどから構成される番組のデータ）が記録されている記録媒体を再生する際、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体からのAVストリームの読み出し位置の決定やストリームの復号といった処理を速やかに行わなくてはならないが、記録媒体に記録されるデータ量が増加するに従い、そのような処理を速やかに行えないといった課題があった。

【0008】本発明はこのような状況に鑑みてなされたものであり、AVストリーム内のIピクチャのアドレス情報、符号化パラメータ、変化点情報、マークなどの情報をファイルとして記録することにより、AVストリームの読み出し位置の決定や復号処理を速やかに行えるようにし、特に、所定のマークを迅速に検索できるようにすることを目的とする。

【0009】

【課題を解決するための手段】本発明の第1の情報処理装置は、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成手段と、生成手段により生成された情報をClip情報として記録媒体に記録する記録手段とを備えることを特徴とする。

【0010】前記符号化情報が連続な区間の開始アドレス情報は、STCシーケンスまたはプログラムシーケンスの開始アドレスであり、時刻情報とアドレス情報を関連

づける情報は、EP_mapまたはTU_mapであり、特徴的な画像の時刻情報は、ClipMarkとすることができる。

【0011】前記記録手段は、AVストリームの記録レートの平均値に関する情報を記録媒体にさらに記録することができる。

【0012】前記平均値に関する情報は、TS_average_rateとすることができる。

【0013】前記AVストリームはトランスポートストリームとすることができる。

【0014】前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのシステムタイムクロックが連続な区間であるSTCシーケンスの開始アドレスを含むようにすることができる。

【0015】前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのプログラム内容が一定な区間であるプログラムシーケンスの開始アドレスを含むようにすることができる。

【0016】前記AVストリーム中の符号化情報が連続な区間の開始アドレス情報は、トランスポートストリームのトランスポートパケットの到着時間に基づくアライバルタイムが連続な区間の開始アドレスを含むようにすることができる。

【0017】前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、トランスポートストリームのIピクチャのアドレスとそのプレゼンテーションタイムスタンプを含むようにすることができる。

【0018】前記AVストリーム中の時刻情報とアドレス情報を関連づける情報の少なくとも一部を圧縮する圧縮手段をさらに備え、記録手段は、圧縮手段により圧縮された情報を記録することができる。

【0019】前記AVストリーム中の時刻情報とアドレス情報を関連づける情報は、トランスポートパケットの到着時刻に基づいたアライバルタイムと、それに対応するトランスポートパケットのAVストリームデータ中のアドレスを含むようにすることができる。

【0020】本発明の第1の情報処理方法は、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを含むことを特徴とする。

【0021】本発明の第1の記録媒体のプログラムは、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを含むことを特徴とする。

【0022】本発明の第1のプログラムは、AVストリー

ム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を生成する生成ステップと、生成ステップにより生成された情報をClip情報として記録媒体に記録する記録ステップとを実行させる。

【0023】本発明の第2の情報処理装置は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生手段と、再生手段により再生されたClip情報に基づいて、AVストリームの出力を制御する制御手段とを備えることを特徴とする。

【0024】本発明の第2の情報処理方法は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを含むことを特徴とする。

【0025】本発明の第2の記録媒体のプログラムは、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを含むことを特徴とする。

【0026】本発明の第2のプログラムは、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報を再生する再生ステップと、再生ステップの処理により再生されたClip情報に基づいて、AVストリームの出力を制御する制御ステップとを実行させるプログラム。

【0027】本発明の記録媒体は、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が記録されていることを特徴とする。

【0028】本発明の第1の情報処理装置および方法、記録媒体、並びにプログラムにおいては、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が記録される。

【0029】本発明の第2の情報処理装置および方法、記録媒体、並びにプログラムにおいては、Clip情報とし

て、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が再生される。

【0030】

【発明の実施の形態】以下に、本発明の実施の形態について、図面を参照して説明する。図1は、本発明を適用した記録再生装置1の内部構成例を示す図である。まず、外部から入力された信号を記録媒体に記録する動作を行う記録部2の構成について説明する。記録再生装置1は、アナログデータ、または、デジタルデータを入力し、記録することができる構成とされている。

【0031】端子11には、アナログのビデオ信号が、端子12には、アナログのオーディオ信号が、それぞれ入力される。端子11に入力されたビデオ信号は、解析部14とAVエンコーダ15に、それぞれ出力される。端子12に入力されたオーディオ信号は、解析部14とAVエンコーダ15に出力される。解析部14は、入力されたビデオ信号とオーディオ信号からシーンチェンジなどの特徴点を抽出する。

【0032】AVエンコーダ15は、入力されたビデオ信号とオーディオ信号を、それぞれ符号化し、符号化ビデオストリーム(V)、符号化オーディオストリーム(A)、およびAV同期等のシステム情報(S)をマルチプレクサ16に出力する。

【0033】符号化ビデオストリームは、例えば、MPEG(Moving Picture Expert Group)2方式により符号化されたビデオストリームであり、符号化オーディオストリームは、例えば、MPEG1方式により符号化されたオーディオストリームや、ドルビーAC3方式(商標)により符号化されたオーディオストリーム等である。マルチプレクサ16は、入力されたビデオおよびオーディオのストリームを、入力システム情報に基づいて多重化して、スイッチ17を介して多重化ストリーム解析部18とソースパケットタイザ19に出力する。

【0034】多重化ストリームは、例えば、MPEG2トランスポートストリームやMPEG2プログラムストリームである。ソースパケットタイザ19は、入力された多重化ストリームを、そのストリームを記録させる記録媒体100のアプリケーションフォーマットに従って、ソースパケットから構成されるAVストリームに符号化する。AVストリームは、ECC(誤り訂正)符号化部20と変調部21でECC符号の付加と変調処理が施され、書き込み部22に出力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100にAVストリームファイルを書き込む(記録する)。

【0035】デジタルインタフェースまたはデジタルテレビジョンチューナから入力されるデジタルテレビジョン放送等のトランスポートストリームは、端子13に入力される。端子13に入力されたトランスポートストリ

ームの記録方式には、2通りあり、それらは、トランスベアレントに記録する方式と、記録ビットレートを下げるなどの目的のために再エンコードをした後に記録する方式である。記録方式の指示情報は、ユーザインターフェースとしての端子24から制御部23へ入力される。

【0036】入力トランスポートストリームをトランスベアレントに記録する場合、端子13に入力されたトランスポートストリームは、スイッチ17を介して多重化ストリーム解析部18と、ソースパケットタイザ19に出力される。これ以降の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0037】入力トランスポートストリームを再エンコードした後に記録する場合、端子13に入力されたトランスポートストリームは、デマルチプレクサ26に入力される。デマルチプレクサ26は、入力されたトランスポートストリームに対してデマルチプレクス処理を施し、ビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)を抽出する。

【0038】デマルチプレクサ26により抽出されたストリーム(情報)のうち、ビデオストリームはAVデコーダ27に、オーディオストリームとシステム情報はマルチプレクサ16に、それぞれ出力される。AVデコーダ27は、入力されたビデオストリームを復号し、その再生ビデオ信号をAVエンコーダ15に出力する。AVエンコーダ15は、入力ビデオ信号を符号化し、符号化ビデオストリーム(V)をマルチプレクサ16に出力する。

【0039】一方、デマルチプレクサ26から出力され、マルチプレクサ16に入力されたオーディオストリームとシステム情報、および、AVエンコーダ15から出力されたビデオストリームは、入力システム情報に基づいて、多重化されて、多重化ストリームとして多重化ストリーム解析部18とソースパケットタイザ19にスイッチ17を介して出力される。これ以後の記録媒体100へAVストリームが記録されるまでの処理は、上述のアナログの入力オーディオ信号とビデオ信号を符号化して記録する場合と同一の処理なので、その説明は省略する。

【0040】本実施の形態の記録再生装置1は、AVストリームのファイルを記録媒体100に記録すると共に、そのファイルを説明するアプリケーションデータベース情報も記録する。アプリケーションデータベース情報は、制御部23により作成される。制御部23への入力情報は、解析部14からの動画像の特徴情報、多重化ストリーム解析部18からのAVストリームの特徴情報、および端子24から入力されるユーザからの指示情報である。

【0041】解析部14から供給される動画像の特徴情報は、AVエンコーダ15がビデオ信号を符号化する場合

において、解析部14により生成されるものである。解析部14は、入力ビデオ信号とオーディオ信号の内容を解析し、入力動画像信号の中の特徴的な画像(クリップマーク)に関係する情報を生成する。これは、例えば、入力ビデオ信号の中のプログラムの開始点、シーンチェンジ点やCMコマーシャルのスタート点・エンド点、タイトルやテロップなどの特徴的なクリップマーク点の画像の指示情報であり、また、それにはその画像のサムネールも含まれる。さらにオーディオ信号のステレオとモノラルの切り換え点や、無音区間などの情報も含まれる。

【0042】これらの画像の指示情報は、制御部23を介して、マルチプレクサ16へ入力される。マルチプレクサ16は、制御部23からクリップマークとして指定される符号化ピクチャを多重化する時に、その符号化ピクチャをAVストリーム上で特定するための情報を制御部23に返す。具体的には、この情報は、ピクチャのPTS(プレゼンテーションタイムスタンプ)またはその符号化ピクチャのAVストリーム上でのアドレス情報である。制御部23は、特徴的な画像の種類とその符号化ピクチャをAVストリーム上で特定するための情報を関連付けて記憶する。

【0043】多重化ストリーム解析部18からのAVストリームの特徴情報は、記録されるAVストリームの符号化情報に関係する情報であり、解析部18により生成される。例えば、AVストリーム内のIピクチャのタイムスタンプとアドレス情報、システムタイムクロックの不連続点情報、AVストリームの符号化パラメータ、AVストリームの中の符号化パラメータの変化点情報などが含まれる。また、端子13から入力されるトランスポートストリームをトランスベアレントに記録する場合、多重化ストリーム解析部18は、入力トランスポートストリームの中から前出のクリップマークの画像を検出し、その種類とクリップマークで指定するピクチャを特定するための情報を生成する。

【0044】端子24からのユーザの指示情報は、AVストリームの中の、ユーザが指定した再生区間の指定情報、その再生区間の内容を説明するキャラクター文字、ユーザが好みのシーンにセットするブックマークやリジューム点の情報などである。

【0045】制御部23は、上記の入力情報に基づいて、AVストリームのデータベース(Clip)、AVストリームの再生区間(PlayItem)をグループ化したもの(PlayList)のデータベース、記録媒体100の記録内容の管理情報(info.dvr)、およびサムネイル画像の情報を作成する。これらの情報から構成されるアプリケーションデータベース情報は、AVストリームと同様に、ECC符号化部20、変調部21で処理されて、書き込み部22へ入力される。書き込み部22は、制御部23から出力される制御信号に基づいて、記録媒体100へデータベースファイルを記録する。

【0046】上述したアプリケーションデータベース情報についての詳細は後述する。

【0047】このようにして記録媒体100に記録されたAVストリームファイル（画像データと音声データのファイル）と、アプリケーションデータベース情報が再生部3により再生される場合、まず、制御部23は、読み出し部28に対して、記録媒体100からアプリケーションデータベース情報を読み出すように指示する。そして、読み出し部28は、記録媒体100からアプリケーションデータベース情報を読み出し、そのアプリケーションデータベース情報は、復調部29とECC復号部30の復調と誤り訂正処理を経て、制御部23へ入力される。

【0048】制御部23は、アプリケーションデータベース情報に基づいて、記録媒体100に記録されているPlayListの一覧を端子24のユーザインターフェースへ出力する。ユーザは、PlayListの一覧から再生したいPlayListを選択し、再生を指定されたPlayListに関する情報が制御部23へ入力される。制御部23は、そのPlayListの再生に必要なAVストリームファイルの読み出しを、読み出し部28に指示する。読み出し部28は、その指示に従い、記録媒体100から対応するAVストリームを読み出し復調部29に出力する。復調部29に入力されたAVストリームは、所定の処理が施されることにより復調され、さらにECC復号部30の処理を経て、ソースデパケッタイザ31出力される。

【0049】ソースデパケッタイザ31は、記録媒体100から読み出され、所定の処理が施されたアプリケーションフォーマットのAVストリームを、デマルチプレクサ26が処理可能なストリームに変換する。デマルチプレクサ26は、制御部23により指定されたAVストリームの再生区間(PlayItem)を構成するビデオストリーム(V)、オーディオストリーム(A)、およびAV同期等のシステム情報(S)を、AVデコーダ27に出力する。AVデコーダ27は、ビデオストリームとオーディオストリームを復号し、再生ビデオ信号と再生オーディオ信号を、それぞれ対応する端子32と端子33から出力する。

【0050】また、ユーザインターフェースとしての端子24から、ランダムアクセス再生や特殊再生を指示する情報が入力された場合、制御部23は、AVストリームのデータベース(Clip)の内容に基づいて、記憶媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを、読み出し部28に指示する。例えば、ユーザにより選択されたPlayListを、所定の時刻から再生する場合、制御部23は、指定された時刻に最も近いタイムスタンプを持つIピクチャからのデータを読み出すように読み出し部28に指示する。

【0051】また、Clip Informationの中のClipMarkにストアされている番組の頭出し点やシーンチェンジ点の中から、ユーザがあるクリップマークを選択した時（例

えば、この動作は、ClipMarkにストアされている番組の頭出し点やシーンチェンジ点のサムネール画像リストをユーザインターフェースに表示して、ユーザが、その中からある画像を選択することにより行われる）、制御部23は、Clip Informationの内容に基づいて、記録媒体100からのAVストリームの読み出し位置を決定し、そのAVストリームの読み出しを読み出し部28へ指示する。すなわち、ユーザが選択した画像がストアされているAVストリーム上でのアドレスに最も近いアドレスにあるIピクチャからのデータを読み出すように読み出し部28へ指示する。読み出し部28は、指定されたアドレスからデータを読み出し、読み出されたデータは、復調部29、ECC復号部30、ソースデパケッタイザ31の処理を経て、デマルチプレクサ26へ入力され、AVデコーダ27で復号されて、マーク点のピクチャのアドレスで示されるAVデータが再生される。

【0052】また、ユーザによって高速再生(Fast-forward playback)が指示された場合、制御部23は、AVストリームのデータベース(Clip)に基づいて、AVストリームの中のI-ピクチャデータを順次連続して読み出すように読み出し部28に指示する。

【0053】読み出し部28は、指定されたランダムアクセスポイントからAVストリームのデータを読み出し、読み出されたデータは、後段の各部の処理を経て再生される。

【0054】次に、ユーザが、記録媒体100に記録されているAVストリームの編集をする場合を説明する。ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合、例えば、番組Aという歌番組から歌手Aの部分を再生し、その後続けて、番組Bという歌番組の歌手Aの部分を再生したいといった再生経路を作成したい場合、ユーザインターフェースとしての端子24から再生区間の開始点（イン点）と終了点（アウト点）の情報が制御部23に入力される。制御部23は、AVストリームの再生区間(PlayItem)をグループ化したもの（PlayList）のデータベースを作成する。

【0055】ユーザが、記録媒体100に記録されているAVストリームの一部を消去したい場合、ユーザインターフェースとしての端子24から消去区間のイン点とアウト点の情報が制御部23に入力される。制御部23は、必要なAVストリーム部分だけを参照するようにPlayListのデータベースを変更する。また、AVストリームの不要なストリーム部分を消去するように、書き込み部22に指示する。

【0056】ユーザが、記録媒体100に記録されているAVストリームの再生区間を指定して新しい再生経路を作成したい場合であり、かつ、それぞれの再生区間をシームレスに接続したい場合について説明する。このような場合、制御部23は、AVストリームの再生区間(Play

tem)をグループ化したもの(Playlist)のデータベースを作成し、さらに、再生区間の接続点付近のビデオストリームの部分的な再エンコードと再多重化を行う。

【0057】まず、端子24から再生区間のイン点のピクチャの情報と、アウト点のピクチャの情報が制御部23へ入力される。制御部23は、読み出し部28にイン点側ピクチャとアウト点側のピクチャを再生するために必要なデータの読み出しを指示する。そして、読み出し部28は、記録媒体100からデータを読み出し、そのデータは、復調部29、ECC復号部30、ソースデパケッタイザ31を経て、デマルチプレクサ26に出力される。

【0058】制御部23は、デマルチプレクサ26に入力されたデータを解析して、ビデオストリームの再エンコード方法(picture_coding_typeの変更、再エンコードする符号化ビット量の割り当て)と、再多重化方式を決定し、その方式をAVエンコーダ15とマルチプレクサ16に供給する。

【0059】次に、デマルチプレクサ26は、入力されたストリームをビデオストリーム(V)、オーディオストリーム(A)、およびシステム情報(S)に分離する。ビデオストリームは、AVデコーダ27に入力されるデータとマルチプレクサ16に入力されるデータがある。前者のデータは、再エンコードするために必要なデータであり、これはAVデコーダ27で復号され、復号されたピクチャはAVエンコーダ15で再エンコードされて、ビデオストリームにされる。後者のデータは、再エンコードをしないで、オリジナルのストリームからコピーされるデータである。オーディオストリーム、システム情報については、直接、マルチプレクサ16に入力される。

【0060】マルチプレクサ16は、制御部23から入力された情報に基づいて、入力ストリームを多重化し、多重化ストリームを出力する。多重化ストリームは、ECC符号化部20、変調部21で処理されて、書き込み部22に入力される。書き込み部22は、制御部23から供給される制御信号に基づいて、記録媒体100にAVストリームを記録する。

【0061】以下に、アプリケーションデータベース情報や、その情報に基づく再生、編集といった操作に関する説明をする。図2は、アプリケーションフォーマットの構造を説明する図である。アプリケーションフォーマットは、AVストリームの管理のためにPlaylistとClipの2つのレイヤをもつ。Volume Informationは、ディスク内のすべてのClipとPlaylistの管理をする。ここでは、1つのAVストリームとその付属情報のペアを1つのオブジェクトと考え、それをClipと称する。AVストリームファイルはClip AV stream fileと称し、その付属情報は、Clip Information fileと称する。

【0062】1つのClip AV stream fileは、MPEG2トランスポートストリームをアプリケーションフォーマット

によって規定される構造に配置したデータをストアする。一般的に、ファイルは、バイト列として扱われるが、Clip AV stream fileのコンテンツは、時間軸上に展開され、Clipの中のエントリーポイント(1ピクチャ)は、主に時間ベースで指定される。所定のClipへのアクセスポイントのタイムスタンプが与えられた時、Clip Information fileは、Clip AV stream fileの中でデータの読み出しを開始すべきアドレス情報を見つけるために役立つ。

【0063】Playlistについて、図3を参照して説明する。Playlistは、Clipの中からユーザが見たい再生区間を選択し、それを簡単に編集することができるようにするために設けられている。1つのPlaylistは、Clipの中の再生区間の集まりである。所定のClipの中の1つの再生区間は、PlayItemと呼ばれ、それは、時間軸上のイン点(IN)とアウト点(OUT)の対で表される。従って、Playlistは、複数のPlayItemが集まることにより構成される。

【0064】Playlistには、2つのタイプがある。1つは、Real Playlistであり、もう1つは、Virtual Playlistである。Real Playlistは、それが参照しているClipのストリーム部分を共有している。すなわち、Real Playlistは、その参照しているClipのストリーム部分に相当するデータ容量をディスクの中で占め、Real Playlistが消去された場合、それが参照しているClipのストリーム部分もまたデータが消去される。

【0065】Virtual Playlistは、Clipのデータを共有していない。従って、Virtual Playlistが変更または消去されたとしても、Clipの内容には何も変化が生じない。

【0066】次に、Real Playlistの編集について説明する。図4(A)は、Real Playlistのクリエイト(create:作成)に関する図であり、AVストリームが新しいClipとして記録される場合、そのClip全体を参照するReal Playlistが新たに作成される操作である。

【0067】図4(B)は、Real Playlistのディバイド(divide:分割)に関する図であり、Real Playlistが所望な点で分けられて、2つのReal Playlistに分割される操作である。この分割という操作は、例えば、1つのPlaylistにより管理される1つのクリップ内に、2つの番組が管理されているような場合に、ユーザが1つ1つの番組として登録(記録)し直したいといったようなときに行われる。この操作により、Clipの内容が変更される(Clip自体が分割される)ことはない。

【0068】図4(C)は、Real Playlistのコンバイン(combine:結合)に関する図であり、2つのReal Playlistを結合して、1つの新しいReal Playlistにする操作である。この結合という操作は、例えば、ユーザが2つの番組を1つの番組として登録し直したいといったようなときに行われる。この操作により、Clipが変更され

る (Clip自体が1つにされる) ことはない。

【0069】図5 (A) は、Real Playlist全体のデリート(delete: 削除)に関する図であり、所定のReal Playlist全体を消去する操作がされた場合、削除されたReal Playlistが参照するClipの、対応するストリーム部分も削除される。

【0070】図5 (B) は、Real Playlistの部分的な削除に関する図であり、Real Playlistの所望な部分が削除された場合、対応するPlayItemが、必要なClipのストリーム部分だけを参照するように変更される。そして、Clipの対応するストリーム部分は削除される。

【0071】図5 (C) は、Real Playlistのミニマイズ(Minimize: 最小化)に関する図であり、Real Playlistに対応するPlayItemを、Virtual Playlistに必要なClipのストリーム部分だけを参照するようにする操作である。Virtual Playlist にとって不必要なClipの、対応するストリーム部分は削除される。

【0072】上述したような操作により、Real Playlistが変更されて、そのReal Playlistが参照するClipのストリーム部分が削除された場合、その削除されたClipを使用しているVirtual Playlistが存在し、そのVirtual Playlistにおいて、削除されたClipにより問題が生じる可能性がある。

【0073】そのようなことが生じないように、ユーザに、削除という操作に対して、「そのReal Playlistが参照しているClipのストリーム部分を参照しているVirtual Playlistが存在し、もし、そのReal Playlistが消去されると、そのVirtual Playlistもまた消去されることになるが、それでも良いか?」といったメッセージなどを表示させることにより、確認(警告)を促した後に、ユーザの指示により削除の処理を実行、または、キャンセルする。または、Virtual Playlistを削除する代わりに、Real Playlistに対してミニマイズの操作が行われるようにする。

【0074】次にVirtual Playlistに対する操作について説明する。Virtual Playlistに対して操作が行われたとしても、Clipの内容が変更されることはない。図6は、アセンブル(Assemble) 編集 (IN-OUT 編集)に関する図であり、ユーザが見たいと所望した再生区間のPlayItemを作り、Virtual Playlistを作成するといった操作である。PlayItem間のシームレス接続が、アプリケーションフォーマットによりサポートされている(後述)。

【0075】図6 (A) に示したように、2つのReal Playlist1, 2と、それぞれのReal Playlistに対応するClip1, 2が存在している場合に、ユーザがReal Playlist1内の所定の区間(In1乃至Out1までの区間: PlayItem1)を再生区間として指示し、続けて再生する区間として、Real Playlist2内の所定の区間(In2乃至Out2までの区間: PlayItem2)を再生区間として指示したとき、図6 (B) に示すように、PlayItem1とPlayItem

2から構成される1つのVirtual Playlistが作成される。

【0076】次に、Virtual Playlist の再編集(Rerediting)について説明する。再編集には、Virtual Playlistの中のイン点やアウト点の変更、Virtual Playlistへの新しいPlayItemの挿入(insert)や追加(append)、Virtual Playlistの中のPlayItemの削除などがある。また、Virtual Playlistそのものを削除することもできる。

【0077】図7は、Virtual Playlistへのオーディオのアフレコ(Audio dubbing (post recording))に関する図であり、Virtual Playlistへのオーディオのアフレコをサブパスとして登録する操作のことである。このオーディオのアフレコは、アプリケーションフォーマットによりサポートされている。Virtual PlaylistのメインパスのAVストリームに、付加的なオーディオストリームが、サブパスとして付加される。

【0078】Real PlaylistとVirtual Playlistで共通の操作として、図8に示すようなPlaylistの再生順序の変更(Moving)がある。この操作は、ディスク(ボリューム)の中でのPlaylistの再生順序の変更であり、アプリケーションフォーマットにおいて定義されるTable Of Playlist (図20などを参照して後述する) によってサポートされる。この操作により、Clipの内容が変更されるようなことはない。

【0079】次に、マーク(Mark)について説明する。マークは、図9に示されるように、ClipおよびPlaylistの中のハイライトや特徴的な時間を指定するために設けられている。Clipに付加されるマークは、ClipMark(クリップマーク)と呼ばれる。ClipMarkは、AVストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。ClipMarkは、図1の例えば解析部14によって生成される。Playlistを再生する時、そのPlaylistが参照するClipのマークを参照して、使用する事ができる。

【0080】Playlistに付加されるマークは、PlaylistMark(プレイリストマーク)と呼ばれる。PlaylistMarkは、主にユーザによってセットされる、例えば、ブックマークやリジューム点などである。ClipまたはPlaylistにマークをセットすることは、マークの時刻を示すタイムスタンプをマークリストに追加することにより行われる。また、マークを削除することは、マークリストの中から、そのマークのタイムスタンプを除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0081】ClipMarkの別のフォーマットとして、ClipMarkが参照するピクチャをAVストリームの中でのアドレススペースで指定するようにしても良い。Clipにマークをセットすることは、マーク点のピクチャを示すアドレスベースの情報をマークリストに追加することにより行われる。また、マークを削除することは、マークリストの

中から、そのマーク点のピクチャを示すアドレススペースの情報を除去する事である。従って、マークの設定や削除により、AVストリームは何の変更もされない。

【0082】次にサムネイルについて説明する。サムネイルは、Volume、PlayList、およびClipに付加される静止画である。サムネイルには、2つの種類があり、1つは、内容を表す代表画としてのサムネイルである。これは主としてユーザがカーソル（不図示）などを操作して見たいものを選択するためのメニュー画面で使われるものである。もう1つは、マークが指しているシーンを表す画像である。

【0083】Volumeと各Playlistは代表画を持つことができるようにする必要がある。Volumeの代表画は、ディスク（記録媒体100、以下、記録媒体100はディスク状のものであるとし、適宜、ディスクと記述する）を記録再生装置1の所定の場所にセットした時に、そのディスクの内容を表す静止画を最初に表示する場合などに用いられることを想定している。Playlistの代表画は、Playlistを選択するメニュー画面において、Playlistの内容を表すための静止画として用いられることを想定している。

【0084】Playlistの代表画として、Playlistの最初の画像をサムネイル（代表画）にすることが考えられるが、必ずしも再生時刻0の先頭の画像が内容を表す上で最適な画像とは限らない。そこで、Playlistのサムネイルとして、任意の画像をユーザが設定できるようにする。以上Volumeを表す代表画としてのサムネイルと、Playlistを表す代表画としてのサムネイルの2種類のサムネイルをメニューサムネイルと称する。メニューサムネイルは頻繁に表示されるため、ディスクから高速に読み出される必要がある。このため、すべてのメニューサムネイルを1つのファイルに格納することが効率的である。メニューサムネイルは、必ずしもボリューム内の動画から抜き出したピクチャである必要はなく、図10に示すように、パーソナルコンピュータやデジタルスチルカメラから取り込まれた画像でもよい。

【0085】一方、ClipとPlaylistには、複数個のマークを打てる必要があり、マーク位置の内容を知るためにマーク点の画像を容易に見ることが出来るようにする必要がある。このようなマーク点を表すピクチャをマークサムネイル（Mark Thumbnails）と称する。従って、マークサムネイルの元となる画像は、外部から取り込んだ画像よりも、マーク点の画像を抜き出したものが主となる。

【0086】図11は、Playlistに付けられるマークと、そのマークサムネイルの関係について示す図であり、図12は、Clipに付けられるマークと、そのマークサムネイルの関係について示す図である。マークサムネイルは、メニューサムネイルと異なり、Playlistの詳細を表す時に、サブメニュー等で使われるため、短いアク

セス時間で読み出されるようなことは要求されない。そのため、サムネイルが必要になる度に、記録再生装置1がファイルを開き、そのファイルの一部を読み出すことで多少時間がかかっても、問題にはならない。

【0087】また、ボリューム内に存在するファイル数を減らすために、すべてのマークサムネイルは1つのファイルに格納するのがよい。Playlistはメニューサムネイル1つと複数のマークサムネイルを有することができ、Clipは直接ユーザが選択する必要性がない（通常、Playlist経由で指定する）ため、メニューサムネイルを設ける必要はない。

【0088】図13は、上述したことを考慮した場合のメニューサムネイル、マークサムネイル、Playlist、およびClipの関係について示した図である。メニューサムネイルファイルには、Playlist毎に設けられたメニューサムネイルがファイルされている。メニューサムネイルファイルには、ディスクに記録されているデータの内容を代表するボリュームサムネイルが含まれている。マークサムネイルファイルは、各Playlist毎と各Clip毎に作成されたサムネイルがファイルされている。

【0089】次に、CPI（Characteristic Point Information）について説明する。CPIは、Clipインフォメーションファイルに含まれるデータであり、主に、それはClipへのアクセスポイントのタイムスタンプが与えられた時、Clip AV stream fileの中でデータの読み出しを開始すべきデータアドレスを見つけるために用いられる。本実施の形態では、2種類のCPIを用いる。1つは、EP_mapであり、もう一つは、TU_mapである。

【0090】EP_mapは、エン트리ポイント（EP）データのリストであり、それはエレメンタリーストリームおよびトランスポートストリームから抽出されたものである。これは、AVストリームの中でデコードを開始すべきエン트리ポイントの場所を見つけるためのアドレス情報を持つ。1つのEPデータは、プレゼンテーションタイムスタンプ（PTS）と、そのPTSに対応するアクセスユニットのAVストリームの中のデータアドレスの対で構成される。

【0091】EP_mapは、主に2つの目的のために使用される。第1に、Playlistの中でプレゼンテーションタイムスタンプによって参照されるアクセスユニットのAVストリームの中のデータアドレスを見つけるために使用される。第2に、ファーストフォワード再生やファーストリバース再生のために使用される。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタックスを解析することができるとき、EP_mapが作成され、ディスクに記録される。

【0092】TU_mapは、デジタルインタフェースを通して入力されるトランスポートパケットの到着時刻に基づいたタイムユニット（TU）データのリストを持つ。これは、到着時刻ベースの時間とAVストリームの中のデータ

アドレスとの関係を与える。記録再生装置1が、入力AVストリームを記録する場合、そのストリームのシンタックスを解析することができないとき、TU_mapが作成され、ディスクに記録される。

【0093】STCInfoは、MPEG2トランスポートストリームをストアしているAVストリームファイルの中にあるSTCの不連続点情報をストアする。

【0094】AVストリームがSTCの不連続点を持つ場合、そのAVストリームファイルの中で同じ値のPTSが現れるかもしれない。そのため、AVストリーム上のある時刻をPTSベースで指す場合、アクセスポイントのPTSだけではそのポイントを特定するためには不十分である。更に、そのPTSを含むところの連続なSTC区間のインデックスが必要である。連続なSTC区間を、このフォーマットではSTC-sequenceと呼び、そのインデックスをSTC-sequence-idと呼ぶ。STC-sequenceの情報は、Clip Information fileのSTCInfoで定義される。

【0095】STC-sequence-idは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【0096】プログラムは、エレメンタリストリームの集まりであり、これらのストリームの同期再生のために、ただ1つのシステムタイムベースを共有するものである。

【0097】再生装置(図1の記録再生装置1)にとって、AVストリームのデコードに先立ち、そのAVストリームの内容がわかることは有用である。例えば、ビデオやオーディオのエレメンタリストリームを伝送するトランスポートパケットのPIDの値や、ビデオやオーディオのコンポーネント種類(例えば、HDTVのビデオとMPEG-2 AACのオーディオストリームなど)などの情報である。この情報はAVストリームを参照するところのPlayListの内容をユーザに説明するメニュー画面を作成するのに有用であるし、また、AVストリームのデコードに先だって、再生装置のAVデコーダおよびデマルチプレクサの初期状態をセットするために役立つ。

【0098】この理由のために、Clip Information fileは、プログラムの内容を説明するためのProgramInfoを持つ。

【0099】MPEG2トランスポートストリームをストアしているAVストリームファイルは、ファイルの中でプログラム内容が変化するかもしれない。例えば、ビデオエレメンタリストリームを伝送するところのトランスポートパケットのPIDが変化したり、ビデオストリームのコンポーネント種類がSDTVからHDTVに変化するなどである。

【0100】ProgramInfoは、AVストリームファイルの中でのプログラム内容の変化点の情報をストアする。AVストリームファイルの中で、このフォーマットで定めるところのプログラム内容が一定である区間をProgram-se-

quenceと呼ぶ。

【0101】Program-sequenceは、EP_mapを持つAVストリームファイルで使用するものであり、TU_mapを持つAVストリームファイルではオプションである。

【0102】本実施の形態では、セルフエンコードのストリームフォーマット(SESF)を定義する。SESFは、アナログ入力信号を符号化する目的、およびデジタル入力信号(例えばDV)をデコードしてからMPEG2トランスポートストリームに符号化する場合に用いられる。

【0103】SESFは、MPEG-2トランスポートストリームおよびAVストリームについてのエレメンタリストリームの符号化制限を定義する。記録再生装置1が、SESFストリームをエンコードし、記録する場合、EP_mapが作成され、ディスクに記録される。

【0104】デジタル放送のストリームは、次に示す方式のうちのいずれかが用いられて記録媒体100に記録される。まず、デジタル放送のストリームをSESFストリームにトランスコーディングする。この場合、記録されたストリームは、SESFに準拠しなければならない。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0105】あるいは、デジタル放送ストリームを構成するエレメンタリストリームを新しいエレメンタリストリームにトランスコーディングし、そのデジタル放送ストリームの規格化組織が定めるストリームフォーマットに準拠した新しいトランスポートストリームに再多重化する。この場合、EP_mapが作成されて、ディスクに記録されなければならない。

【0106】例えば、入力ストリームがISDB(日本のデジタルBS放送の規格名称)準拠のMPEG-2トランスポートストリームであり、それがHDTVビデオストリームとMPEG AACオーディオストリームを含むとする。HDTVビデオストリームをSDTVビデオストリームにトランスコーディングし、そのSDTVビデオストリームとオリジナルのAACオーディオストリームをTSに再多重化する。SDTVストリームと記録されるトランスポートストリームは、共にISDBフォーマットに準拠しなければならない。

【0107】デジタル放送のストリームが、記録媒体100に記録される際の他の方式として、入力トランスポートストリームをトランスペアレントに記録する(入力トランスポートストリームを何も変更しないで記録する)場合であり、その時にEP_mapが作成されてディスクに記録される。

【0108】または、入力トランスポートストリームをトランスペアレントに記録する(入力トランスポートストリームを何も変更しないで記録する)場合であり、その時にTU_mapが作成されてディスクに記録される。

【0109】次にディレクトリとファイルについて説明する。以下、記録再生装置1をDVR(Digital Video Recording)と適宜記述する。図14はディスク上のディレ

クトリ構造の一例を示す図である。DVRのディスク上に必要なディレクトリは、図14に示したように、“DVR”ディレクトリを含むrootディレクトリ、“PLAYLIST”ディレクトリ、“CLIPINF”ディレクトリ、“M2TS”ディレクトリ、および“DATA”ディレクトリを含む“DVR”ディレクトリである。rootディレクトリの下に、これら以外のディレクトリを作成されるようにしても良いが、それらは、本実施の形態のアプリケーションフォーマットでは、無視されるとする。

【0110】“DVR”ディレクトリの下には、DVRアプリケーションフォーマットによって規定される全てのファイルとディレクトリがストアされる。“DVR”ディレクトリは、4個のディレクトリを含む。“PLAYLIST”ディレクトリの下には、Real PlaylistとVirtual Playlistのデータベースファイルが置かれる。このディレクトリは、Playlistが1つなくても存在する。

【0111】“CLIPINF”ディレクトリの下には、Clipのデータベースが置かれる。このディレクトリも、Clipが1つなくても存在する。“M2TS”ディレクトリの下には、AVストリームファイルが置かれる。このディレクトリは、AVストリームファイルが1つなくても存在する。“DATA”ディレクトリは、デジタルTV放送などのデータ放送のファイルがストアされる。

【0112】“DVR”ディレクトリは、次に示すファイルをストアする。“info.dvr”ファイルは、DVRディレクトリの下に作られ、アプリケーションレイヤの全体的な情報をストアする。DVRディレクトリの下には、ただ一つのinfo.dvrがなければならない。ファイル名は、info.dvrに固定されるとする。“menu.thmb”ファイルは、メニューサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのメニューサムネイルがなければならない。ファイル名は、menu.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0113】“mark.thmb”ファイルは、マークサムネイル画像に関連する情報をストアする。DVRディレクトリの下には、ゼロまたは1つのマークサムネイルがなければならない。ファイル名は、mark.thmbに固定されるとする。メニューサムネイル画像が1つもない場合、このファイルは、存在しなくても良い。

【0114】“PLAYLIST”ディレクトリは、2種類のPlaylistファイルをストアするものであり、それらは、Real PlaylistとVirtual Playlistである。“xxxx.rpls”ファイルは、1つのReal Playlistに関連する情報をストアする。それぞれのReal Playlist毎に、1つのファイルが作られる。ファイル名は、“xxxx.rpls”である。ここで、“xxxx”は、5個の0乃至9まで数字である。ファイル拡張子は、“rpls”でなければならないとする。

【0115】“yyyy.vpls”ファイルは、1つのVirtual Playlistに関連する情報をストアする。それぞれのVirtual

Playlist毎に、1つのファイルが作られる。ファイル名は、“yyyy.vpls”である。ここで、“yyyy”は、5個の0乃至9まで数字である。ファイル拡張子は、“vpls”でなければならないとする。

【0116】“CLIPINF”ディレクトリは、それぞれのAVストリームファイルに対応して、1つのファイルをストアする。“zzzzz.clpi”ファイルは、1つのAVストリームファイル(Clip AV stream file または Bridge-Clip AV stream file)に対応するClip Information fileである。ファイル名は、“zzzzz.clpi”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“clpi”でなければならないとする。

【0117】“M2TS”ディレクトリは、AVストリームのファイルをストアする。“zzzzz.m2ts”ファイルは、DVRシステムにより扱われるAVストリームファイルである。これは、Clip AV stream fileまたはBridge-Clip AV streamである。ファイル名は、“zzzzz.m2ts”であり、“zzzzz”は、5個の0乃至9までの数字である。ファイル拡張子は、“m2ts”でなければならないとする。

【0118】“DATA”ディレクトリは、データ放送から伝送されるデータをストアするものであり、データとは、例えば、XML fileやMPEGファイルなどである。

【0119】次に、各ディレクトリ(ファイル)のシンタクスとセマンティクスを説明する。まず、“info.dvr”ファイルについて説明する。図15は、“info.dvr”ファイルのシンタクスを示す図である。“info.dvr”ファイルは、3個のオブジェクトから構成され、それらは、DVRVolume()、TableOfPlayLists()、およびMakersPrivateData()である。

【0120】図15に示したinfo.dvrのシンタクスについて説明するに、TableOfPlayLists_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、TableOfPlayList()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0121】MakersPrivateData_Start_addressは、info.dvrファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word(パディングワード)は、info.dvrのシンタクスに従って挿入される。N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【0122】DVRVolume()は、ボリューム(ディスク)の内容を記述する情報をストアする。図16は、DVRVolume()のシンタクスを示す図である。図16に示したDVRVolume()のシンタクスを説明するに、version_numberは、このDVRVolume()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化される。

【0123】lengthは、このlengthフィールドの直後か

らDVRVolume()の最後までDVRVolume()のバイト数を示す32ビットの符号なし整数で表される。

【0124】ResumeVolume()は、ボリュームの中で最後に再生したReal PlaylistまたはVirtual Playlistのファイル名を記憶している。ただし、Real PlaylistまたはVirtual Playlistの再生をユーザが中断した時の再生位置は、PlaylistMark()において定義されるresume-markにストアされる(図42、図43)。

【0125】図17は、ResumeVolume()のシンタクスを示す図である。図17に示したResumeVolume()のシンタクスを説明するに、valid_flagは、この1ビットのフラグが1にセットされている場合、resume_PlayList_nameフィールドが有効であることを示し、このフラグが0にセットされている場合、resume_PlayList_nameフィールドが無効であることを示す。

【0126】resume_PlayList_nameの10バイトのフィールドは、リジュームされるべきReal PlaylistまたはVirtual Playlistのファイル名を示す。

【0127】図16に示したDVRVolume()のシンタクスのなかの、UIAppInfoVolume は、ボリュームについてのユーザインターフェースアプリケーションのパラメータをストアする。図18は、UIAppInfoVolumeのシンタクスを示す図であり、そのセマンティクスを説明するに、character_setの8ビットのフィールドは、Volume_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0128】name_lengthの8ビットフィールドは、Volume_nameフィールドの中に示されるボリューム名のバイト長を示す。Volume_nameのフィールドは、ボリュームの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはボリュームの名称を示す。Volume_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0129】Volume_protect_flagは、ボリュームの中のコンテンツを、ユーザに制限することなしに見せてよいかどうかを示すフラグである。このフラグが1にセットされている場合、ユーザが正しくPIN番号(パスワード)を入力できたときだけ、そのボリュームのコンテンツを、ユーザに見せる事(再生される事)が許可される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、そのボリュームのコンテンツを、ユーザに見せる事が許可される。

【0130】最初に、ユーザが、ディスクをプレーヤへ挿入した時点において、もしこのフラグが0にセットされているか、または、このフラグが1にセットされていてもユーザがPIN番号を正しく入力できたならば、記録再生装置1は、そのディスクの中のPlaylistの一覧を表示させる。それぞれのPlaylistの再生制限は、volume_p

rotect_flagとは無関係であり、それはUIAppInfoPlaylist()の中に定義されるplayback_control_flagによって示される。

【0131】PINは、4個の0乃至9までの数字で構成され、それぞれの数字は、ISO/IEC 646に従って符号化される。ref_thumbnail_indexのフィールドは、ボリュームに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのボリュームにはサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFF である場合、そのボリュームにはサムネイル画像が付加されていないことを示す。

【0132】次に図15に示したinfo.dvrのシンタクス内のTableOfPlayLists()について説明する。TableOfPlayLists()は、Playlist(Real PlaylistとVirtual Playlist)のファイル名をストアする。ボリュームに記録されているすべてのPlaylistファイルは、TableOfPlayLists()の中に含まれる。TableOfPlayLists()は、ボリュームの中のPlaylistのデフォルトの再生順序を示す。

【0133】図20は、TableOfPlayLists()のシンタクスを示す図であり、そのシンタクスについて説明するに、TableOfPlayListsのversion_numberは、このTableOfPlayListsのバージョンナンバーを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0134】lengthは、このlengthフィールドの直後からTableOfPlayLists()の最後までTableOfPlayLists()のバイト数を示す32ビットの符号なしの整数である。number_of_PlayListsの16ビットのフィールドは、Playlist_file_nameを含むfor-loopのループ回数を示す。この数字は、ボリュームに記録されているPlaylistの数に等しくなければならない。Playlist_file_nameの10バイトの数字は、Playlistのファイル名を示す。

【0135】図21は、TableOfPlayLists()のシンタクスの別の構成を示す図である。図21に示したシンタクスは、図20に示したシンタクスに、UIAppInfoPlaylist(後述)を含ませた構成とされている。このように、UIAppInfoPlaylistを含ませた構成とすることで、TableOfPlayListsを読み出すだけで、メニュー画面を作成することが可能となる。ここでは、図20に示したシンタクスを用いるとして以下の説明をする。

【0136】図15に示したinfo.dvrのシンタクス内のMakersPrivateDataについて説明する。MakersPrivateDataは、記録再生装置1のメーカーが、各社の特別なアプリケーションのために、MakersPrivateData()の中にメーカーのプライベートデータを挿入できるように設けられている。各メーカーのプライベートデータは、それを定義し

たメーカーを識別するために標準化されたmaker_IDを持つ。MakersPrivateData()は、1つ以上のmaker_IDを含んでも良い。

【0137】所定のメーカーが、プライベートデータを挿入したい時に、すでに他のメーカーのプライベートデータがMakersPrivateData()に含まれていた場合、他のメーカーは、既にある古いプライベートデータを消去するのではなく、新しいプライベートデータをMakersPrivateData()の中に追加するようにする。このように、本実施の形態においては、複数のメーカーのプライベートデータが、1つのMakersPrivateData()に含まれることが可能であるようにする。

【0138】図22は、MakersPrivateDataのシンタクスを示す図である。図22に示したMakersPrivateDataのシンタクスについて説明するに、version_numberは、このMakersPrivateData()のバージョンナンバを示す4個のキャラクター文字を示す。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からMakersPrivateData()の最後までMakersPrivateData()のバイト数を示す32ビットの符号なし整数を示す。

【0139】mpd_blocks_start_addressは、MakersPrivateData()の先頭のバイトからの相対バイト数を単位として、最初のmpd_block()の先頭バイトアドレスを示す。相対バイト数はゼロからカウントされる。number_of_maker_entriesは、MakersPrivateData()の中に含まれているメーカープライベートデータのエン트리数を与える16ビットの符号なし整数である。MakersPrivateData()の中に、同じmaker_IDの値を持つメーカープライベートデータが2個以上存在してはならない。

【0140】mpd_block_sizeは、1024バイトを単位として、1つのmpd_blockの大きさを与える16ビットの符号なし整数である。例えば、mpd_block_size=1ならば、それは1つのmpd_blockの大きさが1024バイトであることを示す。number_of_mpd_blocksは、MakersPrivateData()の中に含まれるmpd_blockの数を与える16ビットの符号なし整数である。maker_IDは、そのメーカープライベートデータを作成したDVRシステムの製造メーカーを示す16ビットの符号なし整数である。maker_IDに符号化される値は、このDVRフォーマットのライセンスによって指定される。

【0141】maker_model_codeは、そのメーカープライベートデータを作成したDVRシステムのモデルナンバーコードを示す16ビットの符号なし整数である。maker_model_codeに符号化される値は、このフォーマットのライセンスを受けた製造メーカーによって設定される。start_mpd_block_numberは、そのメーカープライベートデータが開始されるmpd_blockの番号を示す16ビットの符号なし整数である。メーカープライベートデータの先頭データは、mpd_blockの先頭にアラインされなければならない。

い。start_mpd_block_numberは、mpd_blockのfor-loopの中の変数jに対応する。

【0142】mpd_lengthは、バイト単位でメーカープライベートデータの大きさを示す32ビットの符号なし整数である。mpd_blockは、メーカープライベートデータがストアされる領域である。MakersPrivateData()の中のすべてのmpd_blockは、同じサイズでなければならない。

【0143】次に、Real Playlist fileとVirtual Playlist fileについて、換言すれば、xxxxx.rplsとyyyyy.vplsについて説明する。図23は、xxxxx.rpls (Real Playlist)、または、yyyyy.vpls (Virtual Playlist)のシンタクスを示す図である。xxxxx.rplsとyyyyy.vplsは、同一のシンタクス構成をもつ。xxxxx.rplsとyyyyy.vplsは、それぞれ、3個のオブジェクトから構成され、それらは、Playlist()、PlaylistMark()、およびMakersPrivateData()である。

【0144】PlaylistMark_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、PlaylistMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0145】MakersPrivateData_Start_addressは、Playlistファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0146】padding_word (パディングワード) は、Playlistファイルのシンタクスにしたがって挿入され、N1とN2は、ゼロまたは任意の正の整数である。それぞれのパディングワードは、任意の値を取るようにしても良い。

【0147】ここで、既に、簡便に説明したが、Playlistについてさらに説明する。ディスク内にあるすべてのReal Playlistによって、Bridge-Clip (後述)を除くすべてのClipの中の再生区間が参照されていなければならない。かつ、2つ以上のReal Playlistが、それらのPlaylistで示される再生区間を同一のClipの中でオーバーラップさせてはならない。

【0148】図24を参照してさらに説明するに、図24(A)に示したように、全てのClipは、対応するReal Playlistが存在する。この規則は、図24(B)に示したように、編集作業が行われた後においても守られる。従って、全てのClipは、どれかしのReal Playlistを参照することにより、必ず視聴することが可能である。

【0149】図24(C)に示したように、Virtual Playlistの再生区間は、Real Playlistの再生区間またはBridge-Clipの再生区間の中に含まれていなければならない。どのVirtual Playlistにも参照されないBridge-Clipがディスクの中に存在してはならない。

【0150】Real Playlistは、Playlistのリストを含むが、SubPlaylistを含んではならない。Virtual Playlist

istは、PlayItemのリストを含み、PlayList()の中に示されるCPI_typeがEP_map typeであり、かつPlayList_typeが0 (ビデオとオーディオを含むPlayList) である場合、Virtual PlayListは、ひとつのSubPlayItemを含む事ができる。本実施の形態におけるPlayList()では、SubPlayItemはオーディオのアフレコの目的にだけに使用される、そして、1つのVirtual PlayListが持つSubPlayItemの数は、0または1でなければならない。

【0151】次に、PlayListについて説明する。図25は、PlayListのシンタックスを示す図である。図25に示したPlayListのシンタックスを説明するに、version_numberは、このPlayList()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からPlayList()の最後までまでのPlayList()のバイト数を示す32ビットの符号なし整数である。PlayList_typeは、このPlayListのタイプを示す8ビットのフィールドであり、その一例を図26に示す。

【0152】CPI_typeは、1ビットのフラグであり、PlayItem()およびSubPlayItem()によって参照されるClipのCPI_typeの値を示す。1つのPlayListによって参照される全てのClipは、それらのCPI()の中に定義されるCPI_typeの値が同じでなければならない。number_of_PlayItemsは、PlayListの中にあるPlayItemの数を示す16ビットのフィールドである。

【0153】所定のPlayItem()に対応するPlayItem_idは、PlayItem()を含むfor-loopの中で、そのPlayItem()の現れる順番により定義される。PlayItem_idは、0から開始される。number_of_SubPlayItemsは、PlayListの中にあるSubPlayItemの数を示す16ビットのフィールドである。この値は、0または1である。付加的なオーディオストリームのパス(オーディオストリームパス)は、サブパスの一種である。

【0154】次に、図25に示したPlayListのシンタックスのUIAppInfoPlayListについて説明する。UIAppInfoPlayListは、PlayListについてのユーザインターフェースアプリケーションのパラメータをストアする。図27は、UIAppInfoPlayListのシンタックスを示す図である。図27に示したUIAppInfoPlayListのシンタックスを説明するに、character_setは、8ビットのフィールドであり、PlayList_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示したテーブルに準拠する値に対応する。

【0155】name_lengthは、8ビットフィールドであり、PlayList_nameフィールドの中に示されるPlayList名のバイト長を示す。PlayList_nameのフィールドは、PlayListの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはPlayListの名称を示す。PlayList_nameフ

ールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0156】record_time_and_dateは、PlayListが記録された時の日時をストアする56ビットのフィールドである。このフィールドは、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、2001/12/23:01:02:03 は、“0x20011223010203”と符号化される。

【0157】durationは、PlayListの総再生時間を時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、01:45:30は、“0x014530”と符号化される。

【0158】valid_periodは、PlayListが有効である期間を示す32ビットのフィールドである。このフィールドは、8個の数字を4ビットのBinary Coded Decimal (BCD)で符号化したものである。例えば、記録再生装置1は、この有効期間の過ぎたPlayListを自動消去する、といったように用いられる。例えば、2001/05/07 は、“0x20010507”と符号化される。

【0159】maker_idは、そのPlayListを最後に更新したDVRプレーヤ(記録再生装置1)の製造者を示す16ビットの符号なし整数である。maker_idに符号化される値は、DVRフォーマットのライセンスによって割り当てられる。maker_codeは、そのPlayListを最後に更新したDVRプレーヤのモデル番号を示す16ビットの符号なし整数である。maker_codeに符号化される値は、DVRフォーマットのライセンスを受けた製造者によって決められる。

【0160】playback_control_flagのフラグが1にセットされている場合、ユーザが正しくPIN番号を入力してきた場合にだけ、そのPlayListは再生される。このフラグが0にセットされている場合、ユーザがPIN番号を入力しなくても、ユーザは、そのPlayListを視聴することができる。

【0161】write_protect_flagは、図28(A)にテーブルを示すように、1にセットされている場合、write_protect_flagを除いて、そのPlayListの内容は、消去および変更されない。このフラグが0にセットされている場合、ユーザは、そのPlayListを自由に消去および変更できる。このフラグが1にセットされている場合、ユーザが、そのPlayListを消去、編集、または上書きする前に、記録再生装置1はユーザに再確認するようなメッセージを表示させる。

【0162】write_protect_flagが0にセットされているReal PlayListが存在し、かつ、そのReal PlayListのClipを参照するVirtual PlayListが存在し、そのVirtual PlayListのwrite_protect_flagが1にセットされている場合、ユーザが、Real PlayListを消去しようとする場合、記録再生装置1は、そのReal PlayListを消去す

る前に、上記Virtual Playlistの存在をユーザに警告するか、または、そのReal Playlistを"Minimize"する。

【0163】is_played_flagは、図28(B)に示すように、フラグが1にセットされている場合、そのPlaylistは、記録されてから一度は再生されたことを示し、0にセットされている場合、そのPlaylistは、記録されてから一度も再生されたことがないことを示す。

【0164】archiveは、図28(C)に示すように、そのPlaylistがオリジナルであるか、コピーされたものであるかを示す2ビットのフィールドである。ref_thumbnail_indexのフィールドは、Playlistを代表するサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのPlaylistには、Playlistを代表するサムネイル画像が付加されており、そのサムネイル画像は、menu.thumファイルの中にストアされている。その画像は、menu.thumファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのPlaylistには、Playlistを代表するサムネイル画像が付加されていない。

【0165】次にPlayItemについて説明する。1つのPlayItem()は、基本的に次のデータを含む。Clipのファイル名を指定するためのClip_information_file_name、Clipの再生区間を特定するためのIN_timeとOUT_timeのペア、Playlist()において定義されるCPI_typeがEP_map typeである場合、IN_timeとOUT_timeが参照するところのSTC_sequence_id、および、先行するPlayItemと現在のPlayItemとの接続の状態を示すところのconnection_conditionである。

【0166】Playlistが2つ以上のPlayItemから構成される時、それらのPlayItemはPlaylistのグローバル時間軸上に、時間のギャップまたはオーバーラップなしに一列に並べられる。Playlist()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持たない時、そのPlayItemにおいて定義されるIN_timeとOUT_timeのペアは、STC_sequence_idによって指定される同じSTC連続区間上の時間を指していないなければならない。そのような例を図29に示す。

【0167】図30は、Playlist()において定義されるCPI_typeがEP_map typeであり、かつ現在のPlayItemがBridgeSequence()を持つ時、次に説明する規則が適用される場合を示している。現在のPlayItemに先行するPlayItemのIN_time(図の中でIN_time1と示されているもの)は、先行するPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。先行するPlayItemのOUT_time(図の中でOUT_time1と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このOUT_timeは、後述する符号化制限に従っていないなければならない。

【0168】現在のPlayItemのIN_time(図の中でIN_time2と示されているもの)は、現在のPlayItemのBridgeSequenceInfo()の中で指定されるBridge-Clipの中の時間を指している。このIN_timeも、後述する符号化制限に従っていないなければならない。現在のPlayItemのPlayItemのOUT_time(図の中でOUT_time2と示されているもの)は、現在のPlayItemのSTC_sequence_idによって指定されるSTC連続区間上の時間を指している。

【0169】図31に示すように、Playlist()のCPI_typeがTU_map typeである場合、PlayItemのIN_timeとOUT_timeのペアは、同じClip AVストリーム上の時間を指している。

【0170】PlayItemのシンタックスは、図32に示すようになる。図32に示したPlayItemのシンタックスを説明するに、Clip_information_file_nameのフィールドは、ClipInformation fileのファイル名を示す。このClipInformation fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していないなければならない。

【0171】STC_sequence_idは、8ビットのフィールドであり、PlayItemが参照するSTC連続区間のSTC_sequence_idを示す。Playlist()の中で指定されるCPI_typeがTU_map typeである場合、この8ビットフィールドは何も意味を持たず、0にセットされる。IN_timeは、32ビットフィールドであり、PlayItemの再生開始時刻をストアする。IN_timeのセマンティクスは、図33に示すように、Playlist()において定義されるCPI_typeによって異なる。

【0172】OUT_timeは、32ビットフィールドであり、PlayItemの再生終了時刻をストアする。OUT_timeのセマンティクスは、図34に示すように、Playlist()において定義されるCPI_typeによって異なる。

【0173】Connection_Conditionは、図35に示したような先行するPlayItemと、現在のPlayItemとの間の接続状態を示す2ビットのフィールドである。図36は、図35に示したConnection_Conditionの各状態について説明する図である。

【0174】次に、BridgeSequenceInfoについて、図37を参照して説明する。BridgeSequenceInfo()は、現在のPlayItemの付属情報であり、次に示す情報を持つ。Bridge-Clip AV streamファイルとそれに対応するClipInformation file(図45)を指定するBridge_Clip_Information_file_nameを含む。

【0175】また、先行するPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。このアドレスは、RSPN_exit_from_previous_Clipと称される。さらに現在のPlayItemが参照するClip AV stream上のソースパケットのアドレスであり、このソースパケットの前にBr

idge-Clip AV streamファイルの最後のソースパケットが接続される。このアドレスは、RSPN_enter_to_current_Clipと称される。

【0176】図37において、RSPN_arrival_time_discontinuityは、the Bridge-Clip AVstreamファイルの中でアライバルタイムベースの不連続点があるところのソースパケットのアドレスを示す。このアドレスは、Clip Info() (図46) の中において定義される。

【0177】図38は、BridgeSequenceinfoのシンタクスを示す図である。図38に示したBridgeSequenceinfoのシンタクスを説明するに、Bridge_Clip_Information_file_nameのフィールドは、Bridge-Clip AV streamファイルに対応するClip Information fileのファイル名を示す。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、'Bridge-Clip AV stream'を示していなければならない。

【0178】RSPN_exit_from_previous_Clipの32ビットフィールドは、先行するPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットに続いてBridge-Clip AV streamファイルの最初のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、先行するPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0179】RSPN_enter_to_current_Clipの32ビットフィールドは、現在のPlayItemが参照するClip AV stream上のソースパケットの相対アドレスであり、このソースパケットの前にBridge-Clip AV streamファイルの最後のソースパケットが接続される。RSPN_exit_from_previous_Clipは、ソースパケット番号を単位とする大きさであり、現在のPlayItemが参照するClip AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。

【0180】次に、SubPlayItemについて、図39を参照して説明する。SubPlayItem()の使用は、PlayList()のCPI_typeがEP_map typeである場合に許される。本実施の形態においては、SubPlayItemはオーディオのアフレコの目的のためだけに使用されるとする。SubPlayItem()は、次に示すデータを含む。まず、PlayListの中のsub_pathが参照するClipを指定するためのClip_information_file_nameを含む。

【0181】また、Clipの中のsub_pathの再生区間を指定するためのSubPath_IN_timeとSubPath_OUT_timeを含む。さらに、main pathの時間軸上でsub_pathが再生開始する時刻を指定するためのsync_PlayItem_idとsync_start_PTS_of_PlayItemを含む。sub_pathに参照されるオーディオのClip AV streamは、STC不連続点(シス

テムタイムベースの不連続点)を含んではならない。sub_pathに使われるClipのオーディオサンプルのクロックは、main pathのオーディオサンプルのクロックにロックされている。

【0182】図40は、SubPlayItemのシンタクスを示す図である。図40に示したSubPlayItemのシンタクスを説明するに、Clip_Information_file_nameのフィールドは、Clip Information fileのファイル名を示し、それはPlayListの中でsub_pathによって使用される。このClip Information fileのClipInfo()において定義されるClip_stream_typeは、Clip AV streamを示していなければならない。

【0183】SubPath_typeの8ビットのフィールドは、sub_pathのタイプを示す。ここでは、図41に示すように、'0x00'しか設定されておらず、他の値は、将来のために確保されている。

【0184】sync_PlayItem_idの8ビットのフィールドは、main pathの時間軸上でsub_pathが再生開始する時刻が含まれるPlayItemのPlayItem_idを示す。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

【0185】sync_start_PTS_of_PlayItemの32ビットのフィールドは、main pathの時間軸上でsub_pathが再生開始する時刻を示し、sync_PlayItem_idで参照されるPlayItem上のPTS(Presentation Time Stamp)の上位32ビットを示す。SubPath_IN_timeの32ビットフィールドは、Sub_pathの再生開始時刻をストアする。SubPath_IN_timeは、Sub_Pathの中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示す。

【0186】SubPath_OUT_timeの32ビットフィールドは、Sub_pathの再生終了時刻をストアする。SubPath_OUT_timeは、次式によって算出されるPresentation_end_TSの値の上位32ビットを示す。

$$\text{Presentation_end_TS} = \text{PTS_out} + \text{AU_duration}$$

ここで、PTS_outは、SubPathの最後のプレゼンテーションユニットに対応する33ビット長のPTSである。AU_durationは、SubPathの最後のプレゼンテーションユニットの90kHz単位の表示期間である。

【0187】次に、図23に示したxxxxx.rplsとyyyyy.vplsのシンタクス内のPlayListMark()について説明する。PlayListについてのマーク情報は、このPlayListMarkにストアされる。図42は、PlayListMarkのシンタクスを示す図である。図42に示したPlayListMarkのシンタクスについて説明するに、version_numberは、このPlayListMark()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0188】lengthは、このlengthフィールドの直後からPlayListMark()の最後までのPlayListMark()のバイト

数を示す32ビットの符号なし整数である。number_of_PlayList_marksは、PlayListMarkの中にストアされているマークの個数を示す16ビットの符号なし整数である。number_of_PlayList_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図43に示すテーブルに従って符号化される。

【0189】mark_time_stampの32ビットフィールドは、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図44に示すように、PlayList()において定義されるCPI_typeによって異なる。PlayItem_idは、マークが置かれているところのPlayItemを指定する8ビットのフィールドである。所定のPlayItemに対応するPlayItem_idの値は、PlayList()において定義される(図25参照)。

【0190】character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示した値に対応する。name_lengthの8ビットフィールドは、Mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。Mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どのような値が設定されても良い。

【0191】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される(後述)。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない事を示す。

【0192】次に、Clip information fileについて説明する。zzzzz.clpi (Clip information fileファイル)は、図45に示すように6個のオブジェクトから構成される。それらは、ClipInfo()、STC_Info()、ProgramInfo()、CPI()、ClipMark()、およびMakersPrivateData()である。AVストリーム(Clip AVストリームまたはBridge-Clip AV stream)とそれに対応するClip Informationファイルは、同じ数字列の"zzzzz"が使用される。

【0193】図45に示したzzzzz.clpi (Clip information fileファイル)のシンタックスについて説明するに、ClipInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0194】STC_Info_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、STC_Info()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。ProgramInfo_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ProgramInfo()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。CPI_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、CPI()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。

【0195】ClipMark_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、ClipMark()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。MakersPrivateData_Start_addressは、zzzzz.clpiファイルの先頭のバイトからの相対バイト数を単位として、MakersPrivateData()の先頭アドレスを示す。相対バイト数はゼロからカウントされる。padding_word (パディングワード)は、zzzzz.clpiファイルのシンタックスにしたがって挿入される。N1, N2, N3, N4, およびN5は、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値がとられるようにしても良い。

【0196】次に、ClipInfoについて説明する。図46は、ClipInfoのシンタックスを示す図である。ClipInfo()は、それに対応するAVストリームファイル(Clip AVストリームまたはBridge-Clip AVストリームファイル)の属性情報をストアする。

【0197】図46に示したClipInfoのシンタックスについて説明するに、version_numberは、このClipInfo()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。lengthは、このlengthフィールドの直後からClipInfo()の最後までのClipInfo()のバイト数を示す32ビットの符号なし整数である。Clip_stream_typeの8ビットのフィールドは、図47に示すように、Clip Informationファイルに対応するAVストリームのタイプを示す。それぞれのタイプのAVストリームのストリームタイプについては後述する。

【0198】offset_SPNの32ビットのフィールドは、AVストリーム(Clip AVストリームまたはBridge-Clip AVストリーム)ファイルの最初のソースパケットについてのソースパケット番号のオフセット値を与える。AVストリームファイルが最初にディスクに記録される時、このoffset_SPNは0でなければならない。

【0199】図48に示すように、AVストリームファイルのはじめの部分が編集によって消去された時、offset_SPNは、ゼロ以外の値をとっても良い。本実施の形態では、offset_SPNを参照する相対ソースパケット番号(相対アドレス)が、しばしば、RSPN_xxx (xxxは変形する。例、RSPN_EP_start)の形式でシンタックスの中に記

述されている。相対ソースパケット番号は、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからoffset_SPNの値を初期値としてカウントされる。

【0200】AVストリームファイルの最初のソースパケットから相対ソースパケット番号で参照されるソースパケットまでのソースパケットの数 (SPN_xxx) は、次式で算出される。

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}$$

図48に、offset_SPN が、4である場合の例を示す。

【0201】TS_recording_rateは、24ビットの符号なし整数であり、この値は、DVRドライブ (書き込み部22) へまたはDVRドライブ (読み出し部28) からのAVストリームの必要な入出力のビットレートを与える。record_time_and_dateは、Clipに対応するAVストリームが記録された時の日時をストアする56ビットのフィールドであり、年/月/日/時/分/秒について、14個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、2001/12/23:01:02:03は、"0x20011223010203"と符号化される。

【0202】durationは、Clipの総再生時間をアライバルタイムクロックに基づいた時間/分/秒の単位で示した24ビットのフィールドである。このフィールドは、6個の数字を4ビットのBinary Coded Decimal (BCD) で符号化したものである。例えば、01:45:30は、"0x014530"と符号化される。

【0203】time_controlled_flagのフラグは、AVストリームファイルの記録モードを示す。このtime_controlled_flagが1である場合、記録モードは、記録してからの時間経過に対してファイルサイズが比例するようにして記録されるモードであることを示し、次式に示す条件を満たさなければならない。

$$\text{TS_average_rate} \cdot 192/188 \cdot (t - \text{start_time}) - \alpha \leq \text{size_clip}(t)$$

$$< \text{TS_average_rate} \cdot 192/188 \cdot (t - \text{start_time}) + \alpha$$

ここで、TS_average_rateは、AVストリームファイルのトランスポートストリームの平均ビットレートをbytes/second の単位で表したものである。

【0204】また、上式において、tは、秒単位で表される時間を示し、start_timeは、AVストリームファイルの最初のソースパケットが記録された時の時刻であり、秒単位で表される。size_clip(t)は、時刻tにおけるAVストリームファイルのサイズをバイト単位で表したものであり、例えば、start_timeから時刻tまでに10個のソースパケットが記録された場合、size_clip(t)は10・192バイトである。αは、TS_average_rateに依存する定数である。

【0205】time_controlled_flagが0にセットされている場合、記録モードは、記録の時間経過とAVストリームのファイルサイズが比例するように制御していないこ

とを示す。例えば、これは入力トランスポートストリームをトランスペアレント記録する場合である。

【0206】TS_average_rateは、time_controlled_flagが1にセットされている場合、この24ビットのフィールドは、上式で用いているTS_average_rateの値を示す。time_controlled_flagが0にセットされている場合、このフィールドは、何も意味を持たず、0にセットされなければならない。例えば、可変ビットレートのトランスポートストリームは、次に示す手順により符号化される。まずトランスポートレートをTS_recording_rateの値にセットする。次に、ビデオストリームを可変ビットレートで符号化する。そして、ヌルパケットを使用しない事によって、間欠的にトランスポートパケットを符号化する。

【0207】RSPN_arrival_time_discontinuityの32ビットフィールドは、Bridge-Clip AV streamファイル上でアライバルタイムベースの不連続が発生する場所の相対アドレスである。RSPN_arrival_time_discontinuityは、ソースパケット番号を単位とする大きさであり、Bridge-Clip AV streamファイルの最初のソースパケットからClipInfo() において定義されるoffset_SPNの値を初期値としてカウントされる。そのBridge-Clip AV streamファイルの中での絶対アドレスは、上述した

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}$$

に基づいて算出される。

【0208】reserved_for_system_useの144ビットのフィールドは、システム用にリザーブされている。is_format_identifier_validのフラグが1である時、format_identifierのフィールドが有効であることを示す。is_original_network_ID_validのフラグが1である場合、original_network_IDのフィールドが有効であることを示す。is_transport_stream_ID_validのフラグが1である場合、transport_stream_IDのフィールドが有効であることを示す。is_service_ID_validのフラグが1である場合、service_IDのフィールドが有効であることを示す。

【0209】is_country_code_validのフラグが1である時、country_codeのフィールドが有効であることを示す。format_identifierの32ビットフィールドは、トランスポートストリームの中でregistration_descriptor (ISO/IEC13818-1で定義されている) が持つformat_identifierの値を示す。original_network_IDの16ビットフィールドは、トランスポートストリームの中で定義されているoriginal_network_IDの値を示す。transport_stream_IDの16ビットフィールドは、トランスポートストリームの中で定義されているtransport_stream_IDの値を示す。

【0210】service_IDの16ビットフィールドは、トランスポートストリームの中で定義されているservice_IDの値を示す。country_codeの24ビットのフィールド

は、ISO166によって定義されるカントリーコードを示す。それぞれのキャラクター文字は、ISO8859-1で符号化される。例えば、日本は"JPN"と表され、"0x4A 0x500x4E"と符号化される。stream_format_nameは、トランスポートストリームのストリーム定義をしているフォーマット機関の名称を示すISO-646の16個のキャラクターコードである。このフィールドの中の無効なバイトは、値'0xFF'がセットされる。

【0211】format_identifier、original_network_ID、transport_stream_ID、service_ID、country_code、およびstream_format_nameは、トランスポートストリームのサービスプロバイダを示すものであり、これにより、オーディオやビデオストリームの符号化制限、SI(サービスインフォメーション)の規格やオーディオビデオストリーム以外のプライベートデータストリームのストリーム定義を認識することができる。これらの情報は、デコーダが、そのストリームをデコードできるか否か、そしてデコードできる場合にデコード開始前にデコードシステムの初期設定を行うために用いることが可能である。

【0212】次に、STC_Infoについて説明する。ここでは、MPEG-2トランスポートストリームの中でSTCの不連続点(システムタイムベースの不連続点)を含まない時間区間をSTC_sequenceと称し、Clipの中で、STC_sequenceは、STC_sequence_idの値によって特定される。図50は、連続なSTC区間について説明する図である。同じSTC_sequenceの中で同じSTCの値は、決して現れない(ただし、後述するように、Clipの最大時間長は制限されている)。従って、同じSTC_sequenceの中で同じPTSの値もまた、決して現れない。AVストリームが、N(N>0)個のSTC不連続点を含む場合、Clipのシステムタイムベースは、(N+1)個のSTC_sequenceに分割される。

【0213】STC_Infoは、STCの不連続(システムタイムベースの不連続)が発生する場所のアドレスをストアする。図51を参照して説明するように、RSPN_STC_startが、そのアドレスを示し、最後のSTC_sequenceを除くk番目(k>=0)のSTC_sequenceは、k番目のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、(k+1)番目のRSPN_STC_startで参照されるソースパケットが到着した時刻で終わる。最後のSTC_sequenceは、最後のRSPN_STC_startで参照されるソースパケットが到着した時刻から始まり、最後のソースパケットが到着した時刻で終了する。

【0214】図52は、STC_Infoのシンタクスを示す図である。図52に示したSTC_Infoのシンタクスについて説明するに、version_numberは、このSTC_Info()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、"0045"と符号化されなければならない。

【0215】lengthは、このlengthフィールドの直後か

らSTC_Info()の最後までSTC_Info()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map_typeを示す場合、このlengthフィールドはゼロをセットしても良い。CPI()のCPI_typeがEP_map_typeを示す場合、num_of_STC_sequencesは1以上の値でなければならない。

【0216】num_of_STC_sequencesの8ビットの符号なし整数は、Clipの中でのSTC_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。所定のSTC_sequenceに対応するSTC_sequence_idは、RSPN_STC_startを含むfor-loopの中で、そのSTC_sequenceに対応するRSPN_STC_startの現れる順番により定義されるものである。STC_sequence_idは、0から開始される。

【0217】RSPN_STC_startの32ビットフィールドは、AVストリームファイル上でSTC_sequenceが開始するアドレスを示す。RSPN_STC_startは、AVストリームファイルの中でシステムタイムベースの不連続点が発生するアドレスを示す。RSPN_STC_startは、AVストリームの中で新しいシステムタイムベースの最初のPCRを持つソースパケットの相対アドレスとしても良い。RSPN_STC_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip_Info()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV_streamファイルの中での絶対アドレスは、既に上述した

$$SPN_{xxx} = RSPN_{xxx} - offset_SPN$$

により算出される。

【0218】次に、図45に示したzzzzz.clipのシンタクス内のProgramInfoについて説明する。図53を参照しながら説明するに、ここでは、Clipの中で次の特徴をもつ時間区間をprogram_sequenceと呼ぶ。まず、PCR_PIDの値が変わらない。次に、ビデオエレメンタリストリームの数が変化しない。また、それぞれのビデオストリームについてのPIDの値とそのVideoCodingInfoによって定義される符号化情報が変化しない。さらに、オーディオエレメンタリストリームの数が変化しない。また、それぞれのオーディオストリームについてのPIDの値とそのAudioCodingInfoによって定義される符号化情報が変化しない。

【0219】program_sequenceは、同一の時刻において、ただ1つのシステムタイムベースを持つ。program_sequenceは、同一の時刻において、ただ1つのPMTを持つ。ProgramInfo()は、program_sequenceが開始する場所のアドレスをストアする。RSPN_program_sequence_startが、そのアドレスを示す。

【0220】図54は、ProgramInfoのシンタクスを示す図である。図54に示したProgramInfoのシンタクスを説明するに、version_numberは、このProgramInfo()のバージョンナンバーを示す4個のキャラクター文字であ

る。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0221】lengthは、このlengthフィールドの直後からProgramInfo()の最後までProgramInfo()のバイト数を示す32ビットの符号なし整数である。CPI()のCPI_typeがTU_map typeを示す場合、このlengthフィールドはゼロにセットされても良い。CPI()のCPI_typeがEP_map typeを示す場合、number_of_programsは1以上の値でなければならない。

【0222】number_of_program_sequencesの8ビットの符号なし整数は、Clipの中でのprogram_sequenceの数を示す。この値は、このフィールドに続くfor-loopのループ回数を示す。Clipの中でprogram_sequenceが変化しない場合、number_of_program_sequencesは1をセットされなければならない。RSPN_program_sequence_startの32ビットフィールドは、AVストリームファイル上でプログラムシーケンスが開始する場所の相対アドレスである。

【0223】RSPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、

$$\text{SPN}_{xxx} = \text{RSPN}_{xxx} - \text{offset_SPN}$$
 により算出される。シンタックスのfor-loopの中でRSPN_program_sequence_start値は、昇順に現れなければならない。

【0224】PCR_PIDの16ビットフィールドは、そのprogram_sequenceに有効なPCRフィールドを含むトランスポートパケットのPIDを示す。number_of_videosの8ビットフィールドは、video_stream_PIDとVideoCodingInfo()を含むfor-loopのループ回数を示す。number_of_audiosの8ビットフィールドは、audio_stream_PIDとAudioCodingInfo()を含むfor-loopのループ回数を示す。video_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なビデオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くVideoCodingInfo()は、そのvideo_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0225】audio_stream_PIDの16ビットフィールドは、そのprogram_sequenceに有効なオーディオストリームを含むトランスポートパケットのPIDを示す。このフィールドに続くAudioCodingInfo()は、そのaudio_stream_PIDで参照されるビデオストリームの内容を説明しなければならない。

【0226】なお、シンタックスのfor-loopの中でvideo_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でビデオストリームのPIDが符号化されている順番に等しくなければならない。また、シンタク

スのfor-loopの中でaudio_stream_PIDの値の現れる順番は、そのprogram_sequenceに有効なPMTの中でオーディオストリームのPIDが符号化されている順番に等しくなければならない。

【0227】図54は、図54に示したProgramInfo()のシンタックス内のVideoCodingInfo()のシンタックスを示す図である。図55に示したVideoCodingInfo()のシンタックスを説明するに、video_formatの8ビットフィールドは、図56に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオフォーマットを示す。

【0228】frame_rateの8ビットフィールドは、図57に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオのフレームレートを示す。display_aspect_ratioの8ビットフィールドは、図58に示すように、ProgramInfo()の中のvideo_stream_PIDに対応するビデオの表示アスペクト比を示す。

【0229】図59は、図54に示したProgramInfo()のシンタックス内のAudioCodingInfo()のシンタックスを示す図である。図59に示したAudioCodingInfo()のシンタックスを説明するに、audio_codingの8ビットフィールドは、図60に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオの符号化方法を示す。

【0230】audio_component_typeの8ビットフィールドは、図61に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのコンポーネントタイプを示す。sampling_frequencyの8ビットフィールドは、図62に示すように、ProgramInfo()の中のaudio_stream_PIDに対応するオーディオのサンプリング周波数を示す。

【0231】次に、図45に示したzzzzz.clipのシンタックス内のCPI (Characteristic Point Information)について説明する。CPIは、AVストリームの中の時間情報とそのファイルの中のアドレスとを関連づけるためにある。CPIには2つのタイプがあり、それらはEP_mapとTU_mapである。図63に示すように、CPI()の中のCPI_typeがEP_map typeの場合、そのCPI()はEP_mapを含む。図64に示すように、CPI()の中のCPI_typeがTU_map typeの場合、そのCPI()はTU_mapを含む。1つのAVストリームは、1つのEP_mapまたは一つのTU_mapを持つ。AVストリームがSESFトランスポートストリームの場合、それに対応するClipはEP_mapを持たなければならない。

【0232】図65は、CPIのシンタックスを示す図である。図65に示したCPIのシンタックスを説明するに、version_numberは、このCPI()のバージョンナンバを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。lengthは、このlengthフィールドの直後からCPI()の最後までCPI()のバイト数を示す32ビットの符号なし整数である。CPI_typeは、図66に示すように、1ビットのフラグであり、ClipのCPIのタイプを表す。

【0233】次に、図65に示したCPIのシンタクス内のEP_mapについて説明する。EP_mapには、2つのタイプがあり、それはビデオストリーム用のEP_mapとオーディオストリーム用のEP_mapである。EP_mapの中のEP_map_typeが、EP_mapのタイプを区別する。Clipが1つ以上のビデオストリームを含む場合、ビデオストリーム用のEP_mapが使用されなければならない。Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、オーディオストリーム用のEP_mapが使用されなければならない。

【0234】ビデオストリーム用のEP_mapについて図67を参照して説明する。ビデオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、ビデオストリームを送送するトランスポートパケットのPIDを示す。PTS_EP_startは、ビデオストリームのシーケンスヘッダから始まるアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0235】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるビデオストリーム毎に作られる。Clipの中に複数のビデオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0236】オーディオストリーム用のEP_mapは、stream_PID、PTS_EP_start、およびRSPN_EP_startというデータを持つ。stream_PIDは、オーディオストリームを送送するトランスポートパケットのPIDを示す。PTS_EP_startは、オーディオストリームのアクセスユニットのPTSを示す。RSPN_EP_startは、AVストリームの中でPTS_EP_startで参照されるアクセスユニットの第1バイト目を含むソースパケットのアドレスを示す。

【0237】EP_map_for_one_stream_PID()と呼ばれるサブテーブルは、同じPIDを持つトランスポートパケットによって伝送されるオーディオストリーム毎に作られる。Clipの中に複数のオーディオストリームが存在する場合、EP_mapは複数のEP_map_for_one_stream_PID()を含んでも良い。

【0238】EP_mapとSTC_Infoの関係を説明するに、1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較する事により、それぞれのSTC_sequenceに属するEP_mapのデータの境界が分かる(図68を参照)。EP_mapは、同じPIDで伝送される連続したストリームの範囲に対して、1つのEP_map_for_one_stream_PID()を持たねばならない。図69に示したような場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が

連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PID()を持たねばならない。

【0239】図70は、EP_mapのシンタクスを示す図である。図70に示したEP_mapのシンタクスを説明するに、EP_typeは、4ビットのフィールドであり、図71に示すように、EP_mapのエントリーポイントタイプを示す。EP_typeは、このフィールドに続くデータフィールドのセマンティクスを示す。Clipが1つ以上のビデオストリームを含む場合、EP_typeは0('video')にセットされなければならない。または、Clipがビデオストリームを含まず、1つ以上のオーディオストリームを含む場合、EP_typeは1('audio')にセットされなければならない。

【0240】number_of_stream_PIDsの16ビットのフィールドは、EP_map()の中のnumber_of_stream_PIDsを変数にもつfor-loopのループ回数を示す。stream_PID(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるk番目のエレメンタリーストリーム(ビデオまたはオーディオストリーム)を送送するトランスポートパケットのPIDを示す。EP_typeが0('video')に等しい場合、そのエレメンタリーストリームはビデオストリームでなければならない。また、EP_typeが1('audio')に等しい場合、そのエレメンタリーストリームはオーディオストリームでなければならない。

【0241】num_EP_entries(k)の16ビットのフィールドは、EP_map_for_one_stream_PID(num_EP_entries(k))によって参照されるnum_EP_entries(k)を示す。EP_map_for_one_stream_PID_start_address(k)：この32ビットのフィールドは、EP_map()の中でEP_map_for_one_stream_PID(num_EP_entries(k))が始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からの大きさで示される。

【0242】padding_wordは、EP_map()のシンタクスにしたがって挿入されなければならない。XとYは、ゼロまたは任意の正の整数でなければならない。それぞれのパディングワードは、任意の値を取っても良い。

【0243】図72は、EP_map_for_one_stream_PIDのシンタクスを示す図である。図72に示したEP_map_for_one_stream_PIDのシンタクスを説明するに、PTS_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_typeにより異なる。EP_typeが0('video')に等しい場合、このフィールドは、ビデオストリームのシーケンスヘッダで始まるアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。EP_typeが1('audio')に等しい場合、このフィールドは、オーディオストリームのアクセスユニットの33ビット精度のPTSの上位32ビットを持つ。

【0244】RSPN_EP_startの32ビットのフィールドのセマンティクスは、EP_map()において定義されるEP_t

ypeにより異なる。EP_typeが0 ('video')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのシーケンスヘッダの第1バイト目を含むソースパケットの相対アドレスを示す。または、EP_typeが1 ('audio')に等しい場合、このフィールドは、AVストリームの中でPTS_EP_startにより参照されるアクセスユニットのオーディオフレームの第1バイト目を含むソースパケットの相対アドレスを示す。

【0245】RSPN_EP_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAVストリームファイルの中での絶対アドレスは、 $SPN_xxx = RSPN_xxx - offset_SPN$ により算出される。シンタックスのfor-loopの中でRSPN_EP_startの値は、昇順に現れなければならない。

【0246】次に、TU_mapについて、図73を参照して説明する。TU_mapは、ソースパケットのアライバルタイムクロック（到着時刻ベースの時計）に基づいて、1つの時間軸を作る。その時間軸は、TU_map_time_axisと呼ばれる。TU_map_time_axisの原点は、TU_map()の中のoffset_timeによって示される。TU_map_time_axisは、offset_timeから一定の単位に分割される。その単位を、time_unitと称する。

【0247】AVストリームの中の各々のtime_unitの中で、最初の完全な形のソースパケットのAVストリームファイル上のアドレスが、TU_mapにストアされる。これらのアドレスを、RSPN_time_unit_startと称する。TU_map_time_axis上において、k (k>=0)番目のtime_unitが始まる時刻は、TU_start_time(k)と呼ばれる。この値は次式に基づいて算出される。

$TU_start_time(k) = offset_time + k * time_unit_size$
 TU_start_time(k)は、45kHzの精度を持つ。

【0248】図74は、TU_mapのシンタックスを示す図である。図74に示したTU_mapのシンタックスを説明するに、offset_timeの32bit長のフィールドは、TU_map_time_axisに対するオフセットタイムを与える。この値は、Clipの中の最初のtime_unitに対するオフセット時刻を示す。offset_timeは、27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。AVストリームが新しいClipとして記録される場合、offset_timeはゼロにセットされなければならない。

【0249】time_unit_sizeの32ビットフィールドは、time_unitの大きさを与えるものであり、それは27MHz精度のアライバルタイムクロックから導き出される45kHzクロックを単位とする大きさである。time_unit_sizeは、1秒以下 (time_unit_size<=45000) にすることが良い。number_of_time_unit_entriesの32ビット

フィールドは、TU_map()の中にストアされているtime_unitのエントリー数を示す。

【0250】RSPN_time_unit_startの32ビットフィールドは、AVストリームの中でそれぞれのtime_unitが開始する場所の相対アドレスを示す。RSPN_time_unit_startは、ソースパケット番号を単位とする大きさであり、AV streamファイルの最初のソースパケットからClipInfo()において定義されるoffset_SPNの値を初期値としてカウントされる。そのAV streamファイルの中での絶対アドレスは、

$SPN_xxx = RSPN_xxx - offset_SPN$

により算出される。シンタックスのfor-loopの中でRSPN_time_unit_startの値は、昇順に現れなければならない。(k+1)番目のtime_unitの中にソースパケットが何もない場合、(k+1)番目のRSPN_time_unit_startは、k番目のRSPN_time_unit_startと等しくなければならない。

【0251】図45に示したzzzzz.clipのシンタックス内のClipMarkについて説明する。ClipMarkは、クリップについてのマーク情報であり、ClipMarkの中にストアされる。このマークは、記録器（記録再生装置1）によってセットされるものであり、ユーザによってセットされるものではない。

【0252】図75は、ClipMarkのシンタックスを示す図である。図75に示したClipMarkのシンタックスを説明するに、version_numberは、このClipMark()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0253】lengthは、このlengthフィールドの直後からClipMark()の最後までのClipMark()のバイト数を示す32ビットの符号なし整数である。number_of_clip_marksは、ClipMarkの中にストアされているマークの個数を示す16ビットの符号なし整数。number_of_clip_marksは、0であってもよい。mark_typeは、マークのタイプを示す8ビットのフィールドであり、図76に示すテーブルに従って符号化される。

【0254】mark_time_stampは、32ビットフィールドであり、マークが指定されたポイントを示すタイムスタンプをストアする。mark_time_stampのセマンティクスは、図77に示すように、Playlist()の中のCPI_typeにより異なる。

【0255】STC_sequence_idは、CPI()の中のCPI_typeがEP_map_typeを示す場合、この8ビットのフィールドは、mark_time_stampが置かれているところのSTC連続区間のSTC_sequence_idを示す。CPI()の中のCPI_typeがTU_map_typeを示す場合、この8ビットのフィールドは何も意味を持たず、ゼロにセットされる。character_setの8ビットのフィールドは、mark_nameフィールドに符号化されているキャラクター文字の符号化方法を示す。その符号化方法は、図19に示される値に対応する。

【0256】name_lengthの8ビットフィールドは、mark_nameフィールドの中に示されるマーク名のバイト長を示す。mark_nameのフィールドは、マークの名称を示す。このフィールドの中の左からname_length数のバイト数が、有効なキャラクター文字であり、それはマークの名称を示す。mark_nameフィールドの中で、それら有効なキャラクター文字の後の値は、どんな値が入っていても良い。

【0257】ref_thumbnail_indexのフィールドは、マークに付加されるサムネイル画像の情報を示す。ref_thumbnail_indexフィールドが、0xFFFFでない値の場合、そのマークにはサムネイル画像が付加されており、そのサムネイル画像は、mark.thmbファイルの中にストアされている。その画像は、mark.thmbファイルの中でref_thumbnail_indexの値を用いて参照される。ref_thumbnail_indexフィールドが、0xFFFFである場合、そのマークにはサムネイル画像が付加されていない。

【0258】図78は、図75に代わるClipMarkの他のシンタックスを示す図であり、図79は、その場合における、図76に代わるmark_typeのテーブルの例を示す。reserved_for_maker_IDは、mark_typeが、0xC0から0xFFの値を示す時に、そのmark_typeを定義しているメーカーのメーカーIDを示す16ビットのフィールドである。メーカーIDは、DVRフォーマットライセンサーが指定する。mark_entry()は、マーク点に指定されたポイントを示す情報であり、そのシンタックスの詳細は後述する。representative_picture_entry()は、mark_entry()によって示されるマークを代表する画像のポイントを示す情報であり、そのシンタックスの詳細は後述する。

【0259】ClipMarkは、ユーザがAVストリームを再生するときに、その内容を視覚的に検索できるようにするために用いられる。DVRプレーヤーは、GUI(グラフィカルユーザインターフェース)を使用して、ClipMarkの情報をユーザに提示する。ClipMarkの情報を視覚的に表示するためには、mark_entry()が示すピクチャよりもむしろrepresentative_picture_entry()が示すピクチャを示したほうが良い。

【0260】図80に、mark_entry()とrepresentative_picture_entry()の例を示す。例えば、あるプログラムが開始してから、しばらくした後(数秒後)、そのプログラムの番組名(タイトル)が表示されるとする。ClipMarkを作るときは、mark_entry()は、そのプログラムの開始ポイントに置き、representative_picture_entry()は、そのプログラムの番組名(タイトル)が表示されるポイントに置くようにしても良い。

【0261】DVRプレーヤーは、representative_picture_entry()の画像をGUIに表示し、ユーザがその画像を指定すると、DVRプレーヤーは、mark_entry()の置かれたポイントから再生を開始する。

【0262】mark_entry() および representative_pic

ture_entry()のシンタックスを、図81に示す。

【0263】mark_time_stampは、32ビットフィールドであり、mark_entry()の場合はマークが指定されたポイントを示すタイムスタンプをストアし、またrepresentative_picture_entry()の場合、mark_entry()によって示されるマークを代表する画像のポイントを示すタイムスタンプをストアする。

【0264】次に、ClipMarkを指定するために、PTSによるタイムスタンプベースの情報を使用するのではなく、アドレスベースの情報を使用する場合のmark_entry()とrepresentative_picture_entry()のシンタックスの例を図82に示す。

【0265】RSPN_ref_EP_startは、mark_entry()の場合、AVストリームの中でマーク点のピクチャをデコードするためのストリームのエン트리ポイントを示すソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表するピクチャをデコードするためのストリームのエン트리ポイントを示すソースパケットの相対アドレスを示す。RSPN_ref_EP_startの値は、EP_mapの中にRSPN_EP_startとしてストアされていなければならない。かつ、そのRSPN_EP_startに対応するPTS_EP_startの値は、EP_mapの中で、マーク点のピクチャのPTSより過去で最も近い値でなければならない。

【0266】offset_num_picturesは、32ビットのフィールドであり、RSPN_ref_EP_startにより参照されるピクチャから表示順序でマーク点で示されるピクチャまでのオフセットのピクチャ数を示す。この数は、ゼロからカウントされる。図83の例の場合、offset_num_picturesは6となる。

【0267】次に、ClipMarkを指定するために、アドレスベースの情報を使用する場合のmark_entry()とrepresentative_picture_entry()のシンタックスの別の例を図84に示す。

【0268】RSPN_mark_pointは、mark_entry()の場合、AVストリームの中で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。また、representative_picture_entry()の場合、mark_entry()によって示されるマークを代表する符号化ピクチャの第1バイト目を含むソースパケットの相対アドレスを示す。

【0269】RSPN_mark_pointは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義されるoffset_SPNの値を初期値としてカウントされる。

【0270】図85を用いて、ClipMarkとEP_mapの関係を説明する。この例の場合、EP_mapが、エン트리ポイントのアドレスとしてI0、I1、Inを指定しており、これらのアドレスからシーケンスヘッダに続く1ピクチャが

開始しているとする。ClipMarkが、あるマークのアドレスとして、M1を指定している時、そのソースパケットから開始しているピクチャをデコードできるためには、M1のアドレスより前で最も近いエンタリーポイントであるI1からデータを読み出し開始すれば良い。

【0271】MakersPrivateDataについては、図22を参照して既に説明したので、その説明は省略する。

【0272】次に、サムネイルインフォメーション (Thumbnail Information) について説明する。サムネイル画像は、menu.thmbファイルまたはmark.thmbファイルにストアされる。これらのファイルは同じシンタクス構造であり、ただ1つのThumbnail()を持つ。menu.thmbファイルは、メニューサムネイル画像、すなわちVolumeを代表する画像、および、それぞれのPlaylistを代表する画像をストアする。すべてのメニューサムネイルは、ただ1つのmenu.thmbファイルにストアされる。

【0273】mark.thmbファイルは、マークサムネイル画像、すなわちマーク点を表すピクチャをストアする。すべてのPlaylistおよびClipに対するすべてのマークサムネイルは、ただ1つのmark.thmbファイルにストアされる。サムネイルは頻繁に追加、削除されるので、追加操作と部分削除の操作は容易に高速に実行できなければならない。この理由のため、Thumbnail()はブロック構造を有する。画像のデータはいくつかの部分に分割され、各部分は一つのtn_blockに格納される。1つの画像データは連続したtn_blockに格納される。tn_blockの列には、使用されていないtn_blockが存在してもよい。1つのサムネイル画像のバイト長は可変である。

【0274】図86は、menu.thmbとmark.thmbのシンタクスを示す図であり、図87は、図86に示したmenu.thmbとmark.thmbのシンタクス内のThumbnailのシンタクスを示す図である。図87に示したThumbnailのシンタクスについて説明するに、version_numberは、このThumbnail()のバージョンナンバーを示す4個のキャラクター文字である。version_numberは、ISO 646に従って、“0045”と符号化されなければならない。

【0275】lengthは、このlengthフィールドの直後からThumbnail()の最後までMakersPrivateData()のバイト数を示す32ビットの符号なし整数である。tn_block_start_addressは、Thumbnail()の先頭のバイトからの相対バイト数を単位として、最初のtn_blockの先頭バイトアドレスを示す32ビットの符号なし整数である。相対バイト数はゼロからカウントされる。number_of_thumbnailsは、Thumbnail()の中に含まれているサムネイル画像のエンタリー数を与える16ビットの符号なし整数である。

【0276】tn_block_sizeは、1024バイトを単位として、1つのtn_blockの大きさを与える16ビットの符号なし整数である。例えば、tn_block_size=1ならば、それは1つのtn_blockの大きさが1024バイトであることを

示す。number_of_tn_blocksは、このThumbnail()中のtn_blockのエンタリー数を表す16ビットの符号なし整数である。thumbnail_indexは、このthumbnail_indexフィールドから始まるforループ一回分のサムネイル情報で表されるサムネイル画像のインデックス番号を表す16ビットの符号なし整数である。thumbnail_indexとして、0xFFFFという値を使用してはならない。thumbnail_indexはUIAppInfoVolume()、UIAppInfoPlaylist()、PlaylistMark()、およびClipMark()の中のref_thumbnail_indexによって参照される。

【0277】thumbnail_picture_formatは、サムネイル画像のピクチャフォーマットを表す8ビットの符号なし整数で、図88に示すような値をとる。表中のDCFとPNGは“menu.thmb”内でのみ許される。マークサムネイルは、値“0x00” (MPEG-2 Video I-picture)をとらなければならない。

【0278】picture_data_sizeは、サムネイル画像のバイト長をバイト単位で示す32ビットの符号なし整数である。start_tn_block_numberは、サムネイル画像のデータが始まるtn_blockのtn_block番号を表す16ビットの符号なし整数である。サムネイル画像データの先頭は、tb_blockの先頭と一致していなければならない。tn_block番号は、0から始まり、tn_blockのfor-ループ中の変数kの値に関係する。

【0279】x_picture_lengthは、サムネイル画像のフレーム画枠の水平方向のピクセル数を表す16ビットの符号なし整数である。y_picture_lengthは、サムネイル画像のフレーム画枠の垂直方向のピクセル数を表す16ビットの符号なし整数である。tn_blockは、サムネイル画像がストアされる領域である。Thumbnail()の中のすべてのtn_blockは、同じサイズ(固定長)であり、その大きさはtn_block_sizeによって定義される。

【0280】図89は、サムネイル画像データがどのようにtn_blockに格納されるかを模式的に表した図である。図89のように、各サムネイル画像データはtn_blockの先頭から始まり、1 tn_blockを超える大きさの場合は、連続する次のtn_blockを使用してストアされる。このようにすることにより、可変長であるピクチャデータが、固定長のデータとして管理することが可能となり、削除といった編集に対して簡便な処理により対応する事ができるようになる。

【0281】次に、AVストリームファイルについて説明する。AVストリームファイルは、“M2TS”ディレクトリ(図14)にストアされる。AVストリームファイルには、2つのタイプがあり、それらは、Clip AVストリームとBridge-Clip AVストリームファイルである。両方のAVストリーム共に、これ以降で定義されるDVR MPEG-2トランスポートストリームファイルの構造でなければならない。

【0282】まず、DVR MPEG-2 トランスポートストリ

ームについて説明する。DVR MPEG-2トランスポートストリームの構造は、図90に示すようになっている。AVストリームファイルは、DVR MPEG2トランスポートストリームの構造を持つ。DVR MPEG2トランスポートストリームは、整数個のAligned unitから構成される。Aligned unitの大きさは、6144 バイト (2048*3 バイト)である。Aligned unitは、ソースパケットの第1バイト目から始まる。ソースパケットは、192バイト長である。一つのソースパケットは、TP_extra_headerとトランスポートパケットから成る。TP_extra_headerは、4バイト長であり、またトランスポートパケットは、188バイト長である。

【0283】1つのAligned unitは、32個のソースパケットから成る。DVR MPEG2トランスポートストリームの中の最後のAligned unitも、また32個のソースパケットから成る。よって、DVR MPEG2トランスポートストリームは、Aligned unitの境界で終端する。ディスクに記録される入力トランスポートストリームのトランスポートパケットの数が32の倍数でない時、ヌルパケット(PID=0x1FFFのトランスポートパケット)を持ったソースパケットを最後のAligned unitに使用しなければならない。ファイルシステムは、DVR MPEG2トランスポートストリームに余分な情報を付加してはならない。

【0284】図91に、DVR MPEG-2トランスポートストリームのレコーダモデルを示す。図91に示したレコーダは、レコーディングプロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【0285】MPEG-2トランスポートストリームの入力タイミングについて説明する。入力MPEG2トランスポートストリームは、フルトランスポートストリームまたはパーシャルトランスポートストリームである。入力されるMPEG2トランスポートストリームは、ISO/IEC13818-1またはISO/IEC13818-9に従っていなければならない。MPEG2トランスポートストリームのi番目のバイトは、T-STD (ISO/IEC 13818-1で規定されるTransport stream system target decoder) 51とソースパケットタイザ(source packetizer) 54へ、時刻t(i)に同時に入力される。Rp kは、トランスポートパケットの入力レートの瞬時的な最大値である。

【0286】27MHz PLL 52は、27MHzクロックの周波数を発生する。27MHzクロックの周波数は、MPEG-2トランスポートストリームのPCR (Program Clock Reference)の値にロックされる。アライバルタイムクロックカウンタ(arrival time clock counter) 53は、27MHzの周波数のパルスをカウントするバイナリーカウンタである。Arrival_time_clock(i)は、時刻t(i)におけるarrival time clockcounter 53のカウント値である。

【0287】source packetizer 54は、すべてのトラ

ンスポートパケットにTP_extra_headerを付加し、ソースパケットを作る。Arrival_time_stampは、トランスポートパケットの第1バイト目がT-STD 51とソースパケットタイザ54の両方へ到着する時刻を表す。Arrival_time_stamp(k)は、次式で示されるようにArrival_time_clock(k)のサンプル値であり、ここで、kはトランスポートパケットの第1バイト目を示す。

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \cdot 2^{30}$$

【0288】2つの連続して入力されるトランスポートパケットの時間間隔が、 $2^{30}/27000000$ 秒(約40秒)以上になる場合、その2つのトランスポートパケットのarrival_time_stampの差分は、 $2^{30}/27000000$ 秒になるようにセットされるべきである。レコーダは、そのようになる場合に備えてある。

【0289】スムージングバッファ(smoothing buffer) 55は、入力トランスポートストリームのビットレートをスムージングする。スムージングバッファ55は、オーバーフローしてはならない。Rmaxは、スムージングバッファ55が空でない時のスムージングバッファ55からのソースパケットの出力ビットレートである。スムージングバッファ55が空である時、スムージングバッファ55からの出力ビットレートはゼロである。

【0290】次に、DVR MPEG-2トランスポートストリームのレコーダモデルのパラメータについて説明する。Rmaxという値は、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateによって与えられる。この値は、次式により算出される。

$$R_{\max} = \text{TS_recording_rate} \cdot 192/188$$

TS_recording_rateの値は、bytes/secondを単位とする大きさである。

【0291】入力トランスポートストリームがSESFトランスポートストリームの場合、Rp kは、AVストリームファイルに対応するClipInfo()において定義されるTS_recording_rateに等しくなければならない。入力トランスポートストリームがSESFトランスポートストリームでない場合、この値はMPEG-2 transport streamのデスクリプター、例えばmaximum_bitrate_descriptorやpartial_transport_stream_descriptorなど、において定義される値を参照しても良い。

【0292】入力トランスポートストリームがSESFトランスポートストリームの場合、スムージングバッファ55の大きさ(smoothing buffer size)はゼロである。入力トランスポートストリームがSESFトランスポートストリームでない場合、スムージングバッファ55の大きさはMPEG-2 transport streamのデスクリプター、例えばsmoothing_buffer_descriptor、short_smoothing_buffer_descriptor、partial_transport_stream_descriptorなどにおいて定義される値を参照しても良い。

【0293】記録機(レコーダ)および再生機(プレーヤ)は、十分なサイズのバッファを用意しなければなら

ない。デフォルトのバッファサイズは、1536 bytesである。

【0294】次に、DVR MPEG-2トランスポートストリームのプレーヤモデルについて説明する。図92は、DVR MPEG-2トランスポートストリームのプレーヤモデルを示す図である。これは、再生プロセスを規定するための概念上のモデルである。DVR MPEG-2トランスポートストリームは、このモデルに従う。

【0295】27MHz X-tal (クリスタル発振器) 61は、27MHzの周波数を発生する。27MHz周波数の誤差範囲は、 ± 30 ppm (27000000 \pm 810 Hz)でなければならない。arrival time clock counter 62は、27MHzの周波数のパルスをカウントするバイナリーカウンタである。arrival_time_clock(i)は、時刻*t*(i)におけるarrival time clock counter 62のカウント値である。

【0296】smoothing buffer 64において、Rmaxは、スムージングバッファ64がフルでない時のスムージングバッファ64へのソースパケットの入力ビットレートである。スムージングバッファ64がフルである時、スムージングバッファ64への入力ビットレートはゼロである。

【0297】MPEG-2トランスポートストリームの出力タイミングを説明するに、現在のソースパケットのarrival_time_stampがarrival_time_clock(i)のLSB 30ビットの値と等しい時、そのソースパケットのトランスポートパケットは、スムージングバッファ64から引き抜かれる。Rpkは、トランスポートパケットレートの瞬時的な最大値である。スムージングバッファ64は、アンダーフローしてはならない。

【0298】DVR MPEG-2トランスポートストリームのプレーヤモデルのパラメータについては、上述したDVR MPEG-2トランスポートストリームのレコーダモデルのパラメータと同一である。

【0299】図93は、Source packetのシンタクスを示す図である。transport_packet()は、ISO/IEC 13818-1で規定されるMPEG-2トランスポートパケットである。図93に示したSource packetのシンタクス内のTP_Extra_headerのシンタクスを図94に示す。図94に示したTP_Extra_headerのシンタクスについて説明するに、copy_permission_indicatorは、トランスポートパケットのペイロードのコピー制限を表す整数である。コピー制限は、copy free、no more copy、copy once、またはcopy prohibitedとすることができる。図95は、copy_permission_indicatorの値と、それらによって指定されるモードの関係を示す。

【0300】copy_permission_indicatorは、すべてのトランスポートパケットに付加される。IEEE1394デジタルインターフェースを使用して入力トランスポートストリームを記録する場合、copy_permission_indicatorの

値は、IEEE1394 isochronous packet headerの中のEMI (Encryption Mode Indicator)の値に関連付けても良い。IEEE1394デジタルインターフェースを使用しないで入力トランスポートストリームを記録する場合、copy_permission_indicatorの値は、トランスポートパケットの中に埋め込まれたCCIの値に関連付けても良い。アナログ信号入力をセルフエンコードする場合、copy_permission_indicatorの値は、アナログ信号のCGMS-Aの値に関連付けても良い。

【0301】arrival_time_stampは、次式

$$\text{arrival_time_stamp}(k) = \text{arrival_time_clock}(k) \% 2^{30}$$
 において、arrival_time_stampによって指定される値を持つ整数値である。

【0302】Clip AVストリームの定義をするに、Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。arrival_time_clock(i)は、Clip AVストリームの中で連続して増加しなければならない。Clip AVストリームの中にシステムタイムベース (STCベース) の不連続点が存在したとしても、そのClip AVストリームのarrival_time_clock(i)は、連続して増加しなければならない。

【0303】Clip AVストリームの中の開始と終了の間のarrival_time_clock(i)の差分の最大値は、26時間でなければならない。この制限は、MPEG2トランスポートストリームの中にシステムタイムベース (STCベース) の不連続点が存在しない場合に、Clip AVストリームの中で同じ値のPTS (Presentation Time Stamp) が決して現れないことを保証する。MPEG2システムズ規格は、PTSのラップアラウンド周期を233/90000秒 (約26.5時間) と規定している。

【0304】Bridge-Clip AVストリームの定義をするに、Bridge-Clip AVストリームは、上述したような定義がされるDVR MPEG-2トランスポートストリームの構造を持たねばならない。Bridge-Clip AVストリームは、1つのアライバルタイムベースの不連続点を含まなければならない。アライバルタイムベースの不連続点の前後のトランスポートストリームは、後述する符号化の制限に従わなければならない、かつ後述するDVR-STDに従わなければならない。

【0305】本実施の形態においては、編集におけるPlayItem間のビデオとオーディオのシームレス接続をサポートする。PlayItem間をシームレス接続にすることは、プレーヤ/レコーダに"データの連続供給"と"シームレスな復号処理"を保証する。"データの連続供給"とは、ファイルシステムが、デコーダにバッファのアンダーフローを起こさせる事のないように必要なビットレートでデータを供給する事を保証できることである。データのリアルタイム性を保証して、データをディスクから読み出すことができるように、データが十分な大きさの連続

したブロック単位でストアされるようにする。

【0306】"シームレスな復号処理"とは、プレーヤが、デコーダの再生出力にポーズやギャップを起こさせることなく、ディスクに記録されたオーディオビデオデータを表示できることである。

【0307】シームレス接続されているPlayItemが参照するAVストリームについて説明する。先行するPlayItemと現在のPlayItemの接続が、シームレス表示できるように保証されているかどうかは、現在のPlayItemにおいて定義されているconnection_conditionフィールドから判断することができる。PlayItem間のシームレス接続は、Bridge-Clipを使用する方法と使用しない方法がある。

【0308】図96は、Bridge-Clipを使用する場合の先行するPlayItemと現在のPlayItemの関係を示している。図96においては、プレーヤが読み出すストリームデータが、影をつけて示されている。図96に示したTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータから成る。

【0309】TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図96においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから、RSPN_exit_from_previous_Clipで参照されるソースパケットまでのストリームデータである。TS1に含まれるBridge-ClipのRSPN_arrival_time_discontinuityより前の影を付けられたストリームデータは、Bridge-Clipの最初のソースパケットから、RSPN_arrival_time_discontinuityで参照されるソースパケットの直前のソースパケットまでのストリームデータである。

【0310】また、図96におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータとBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータから成る。TS2に含まれるBridge-ClipのRSPN_arrival_time_discontinuity以後の影を付けられたストリームデータは、RSPN_arrival_time_discontinuityで参照されるソースパケットから、Bridge-Clipの最後のソースパケットまでのストリームデータである。TS2のClip2の影を付けられたストリームデータは、RSPN_enter_to_current_Clipで参照されるソースパケットから、現在のPlayItemのOUT_time (図96においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0311】図97は、Bridge-Clipを使用しない場合の先行するPlayItemと現在のPlayItemの関係を示している。この場合、プレーヤが読み出すストリームデータは、影をつけて示されている。図97におけるTS1は、Clip1 (Clip AVストリーム) の影を付けられたストリーム

データから成る。TS1のClip1の影を付けられたストリームデータは、先行するPlayItemのIN_time (図97においてIN_time1で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスから始まり、Clip1の最後のソースパケットまでのデータである。また、図97におけるTS2は、Clip2 (Clip AVストリーム) の影を付けられたストリームデータから成る。

【0312】TS2のClip2の影を付けられたストリームデータは、Clip2の最初のソースパケットから始まり、現在のPlayItemのOUT_time (図97においてOUT_time2で図示されている) に対応するプレゼンテーションユニットを復号する為に必要なストリームのアドレスまでのストリームデータである。

【0313】図96と図97において、TS1とTS2は、ソースパケットの連続したストリームである。次に、TS1とTS2のストリーム規定と、それらの間の接続条件について考える。まず、シームレス接続のための符号化制限について考える。トランスポートストリームの符号化構造の制限として、まず、TS1とTS2の中に含まれるプログラムの数は、1でなければならない。TS1とTS2の中に含まれるビデオストリームの数は、1でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、2以下でなければならない。TS1とTS2の中に含まれるオーディオストリームの数は、等しくなければならない。TS1および/またはTS2の中に、上記以外のエレメンタリーストリームまたはプライベートストリームが含まれていても良い。

【0314】ビデオビットストリームの制限について説明する。図98は、ピクチャの表示順序で示すシームレス接続の例を示す図である。接続点においてビデオストリームをシームレスに表示できるように、OUT_time1 (Clip1のOUT_time) の後とIN_time2 (Clip2のIN_time) の前に表示される不必要なピクチャは、接続点付近のClipの部分的なストリームを再エンコードするプロセスにより、除去されなければならない。

【0315】図98に示したような場合において、BridgeSequenceを使用してシームレス接続を実現する例を、図99に示す。RSPN_arrival_time_discontinuityより前のBridge-Clipのビデオストリームは、図98のClip1のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成る。そして、そのビデオストリームは先行するClip1のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリーストリームとなるように再エンコードされている。

【0316】同様にして、RSPN_arrival_time_discontinuity以後のBridge-Clipのビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成る。そして、そのビデオストリームは、正しくデコード開始する事ができて、これに続く

Clip2のビデオストリームに接続され、1つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。Bridge-Clipを作るためには、一般に、数枚のピクチャは再エンコードしなければならない、それ以外のピクチャはオリジナルのClipからコピーすることができる。

【0317】図98に示した例の場合にBridgeSequenceを使用しないでシームレス接続を実現する例を図100に示す。Clip1のビデオストリームは、図98のOUT_time1に対応するピクチャまでの符号化ビデオストリームから成り、それは、1つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。同様に、Clip2のビデオストリームは、図98のClip2のIN_time2に対応するピクチャ以後の符号化ビデオストリームから成り、それは、一つの連続でMPEG2規格に従ったエレメンタリストリームとなるように再エンコードされている。

【0318】ビデオストリームの符号化制限について説明するに、まず、TS1とTS2のビデオストリームのフレームレートは、等しくなければならない。TS1のビデオストリームは、sequence_end_codeで終端しなければならない。TS2のビデオストリームは、Sequence Header、GOP Header、そしてI-ピクチャで開始しなければならない。TS2のビデオストリームは、クローズドGOPで開始しなければならない。

【0319】ビットストリームの中で定義されるビデオプレゼンテーションユニット（フレームまたはフィールド）は、接続点を挟んで連続でなければならない。接続点において、フレームまたはフィールドのギャップがあってはならない。接続点において、トップ?ボトム?のフィールドシーケンスは連続でなければならない。3-2フルダウンを使用するエンコードの場合は、"top_field_first" および "repeat_first_field" フラグを書き換える必要があるかもしれない、またはフィールドギャップの発生を防ぐために局所的に再エンコードするようにしても良い。

【0320】オーディオビットストリームの符号化制限について説明するに、TS1とTS2のオーディオのサンプリング周波数は、同じでなければならない。TS1とTS2のオーディオの符号化方法（例、MPEG1レイヤ2、AC-3、SESLPCM、AAC）は、同じでなければならない。

【0321】次に、MPEG-2トランスポートストリームの符号化制限について説明するに、TS1のオーディオストリームの最後のオーディオフレームは、TS1の最後の表示ピクチャの表示終了時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。TS2のオーディオストリームの最初のオーディオフレームは、TS2の最初の表示ピクチャの表示開始時に等しい表示時刻を持つオーディオサンプルを含んでいなければならない。

【0322】接続点において、オーディオプレゼンテ-

ションユニットのシーケンスにギャップがあってはならない。図101に示すように、2オーディオフレーム区間未満のオーディオプレゼンテーションユニットの長さで定義されるオーバーラップがあっても良い。TS2のエレメンタリストリームを伝送する最初のパケットは、ビデオパケットでなければならない。接続点におけるトランスポートストリームは、後述するDVR-STDに従わなくてはならない。

【0323】ClipおよびBridge-Clipの制限について説明するに、TS1とTS2は、それぞれの中にアライバルタイムベースの不連続点を含んではならない。

【0324】以下の制限は、Bridge-Clipを使用する場合にのみ適用される。TS1の最後のソースパケットとTS2の最初のソースパケットの接続点においてのみ、Bridge-ClipAVストリームは、ただ1つのアライバルタイムベースの不連続点を持つ。ClipInfo()において定義されるRSPN_arrival_time_discontinuityが、その不連続点のアドレスを示し、それはTS2の最初のソースパケットを参照するアドレスを示さなければならない。

【0325】BridgeSequenceInfo()において定義されるRSPN_exit_from_previous_Clipによって参照されるソースパケットは、Clip1の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。BridgeSequenceInfo()において定義されるRSPN_enter_to_current_Clipによって参照されるソースパケットは、Clip2の中のどのソースパケットでも良い。それは、Aligned unitの境界である必要はない。

【0326】PlayItemの制限について説明するに、先行するPlayItemのOUT_time（図96、図97において示されるOUT_time1）は、TS1の最後のビデオプレゼンテーションユニットの表示終了時刻を示さなければならない。現在のPlayItemのIN_time（図96、図97において示されるIN_time2）は、TS2の最初のビデオプレゼンテーションユニットの表示開始時刻を示さなければならない。

【0327】Bridge-Clipを使用する場合のデータアロケーションの制限について、図102を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1（ClipAVストリームファイル）とClip2（ClipAVストリームファイル）に接続されるBridge-ClipAVストリームを、データアロケーション規定を満たすように配置することによって行われなければならない。

【0328】RSPN_exit_from_previous_Clip以前のClip1（ClipAVストリームファイル）のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_exit_from_previous_Clipが選択されなければならない。Bridge-ClipAVストリームのデータ長は、ハーフフラグメント以上の連続領域に配置されるよう

に、選択されなければならない。RSPN_enter_to_current_Clip以後のClip2 (Clip AVストリームファイル) のストリーム部分が、ハーフフラグメント以上の連続領域に配置されているように、RSPN_enter_to_current_Clipが選択されなければならない。

【0329】Bridge-Clipを使用しないでシームレス接続する場合のデータロケーションの制限について、図103を参照して説明するに、シームレス接続は、ファイルシステムによってデータの連続供給が保証されるように作られなければならない。これは、Clip1 (Clip AVストリームファイル) の最後の部分とClip2 (Clip AVストリームファイル) の最初の部分を、データロケーション規定を満たすように配置することによって行われなければならない。

【0330】Clip1 (Clip AVストリームファイル) の最後のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。Clip2 (Clip AVストリームファイル) の最初のストリーム部分が、ハーフフラグメント以上の連続領域に配置されていなければならない。

【0331】次に、DVR-STDについて説明する。DVR-STDは、DVR MPEG2トランスポートストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルである。また、DVR-STDは、上述したシームレス接続された2つのPlayItemによって参照されるAVストリームの生成および検証の際におけるデコード処理をモデル化するための概念モデルでもある。

【0332】DVR-STDモデルを図104に示す。図104に示したモデルには、DVR MPEG-2トランスポートストリームプレーヤモデルが構成要素として含まれている。n, TBn, MBn, EBn, TBSys, Bsys, Rxn, Rbxn, Rxsys, Dn, Dsys, OnおよびPn(k)の表記方法は、ISO/IEC13818-1のT-STDに定義されているものと同じである。すなわち、次の通りである。nは、エレメンタリーストリームのインデックス番号である。TBnは、エレメンタリーストリームnのトランスポートバッファである。

【0333】MBnは、エレメンタリーストリームnの多重バッファである。ビデオストリームについてのみ存在する。EBnは、エレメンタリーストリームnのエレメンタリーストリームバッファである。ビデオストリームについてのみ存在する。TBSysは、復号中のプログラムのシステム情報のための入力バッファである。Bsysは、復号中のプログラムのシステム情報のためのシステムターゲットデコーダ内のメインバッファである。Rxnは、データがTBnから取り除かれる伝送レートである。Rbxnは、PESパケットペイロードがMBnから取り除かれる伝送レートである。ビデオストリームについてのみ存在する。

【0334】Rxsysは、データがTBSysから取り除かれる伝送レートである。Dnは、エレメンタリーストリームnのデコーダである。Dsysは、復号中のプログラムのシス

テム情報に関するデコーダである。Onは、ビデオストリームnのre-ordering bufferである。Pn(k)は、エレメンタリーストリームnのk番目のプレゼンテーションユニットである。

【0335】DVR-STDのデコーディングプロセスについて説明する。単一のDVR MPEG-2トランスポートストリームを再生している間は、トランスポートパケットをTB1, TBnまたはTBSysのバッファへ入力するタイミングは、ソースパケットのarrival_time_stampにより決定される。TB1, MB1, EB1, TBn, Bn, TBSysおよびBsysのバッファリング動作の規定は、ISO/IEC 13818-1に規定されているT-STDと同じである。復号動作と表示動作の規定もまた、ISO/IEC 13818-1に規定されているT-STDと同じである。

【0336】シームレス接続されたPlayItemを再生している間のデコーディングプロセスについて説明する。ここでは、シームレス接続されたPlayItemによって参照される2つのAVストリームの再生について説明をすることにし、以後の説明では、上述した(例えば、図96に示した)TS1とTS2の再生について説明する。TS1は、先行するストリームであり、TS2は、現在のストリームである。

【0337】図105は、あるAVストリーム(TS1)からそれにシームレスに接続された次のAVストリーム(TS2)へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャートを示す。所定のAVストリーム(TS1)からそれにシームレスに接続された次のAVストリーム(TS2)へと移る間には、TS2のアライバルタイムベースの時間軸(図105においてATC2で示される)は、TS1のアライバルタイムベースの時間軸(図105においてATC1で示される)と同じでない。

【0338】また、TS2のシステムタイムベースの時間軸(図105においてSTC2で示される)は、TS1のシステムタイムベースの時間軸(図105においてSTC1で示される)と同じでない。ビデオの表示は、シームレスに連続していることが要求される。オーディオのプレゼンテーションユニットの表示時間にはオーバーラップがあっても良い。

【0339】DVR-STD への入力タイミングについて説明する。時刻 T_1 までの時間、すなわち、TS1の最後のビデオパケットがDVR-STDのTB1に入力終了するまでは、DVR-STDのTB1, TBn またはTBSysのバッファへの入力タイミングは、TS1のソースパケットのarrival_time_stampによって決定される。

【0340】TS1の残りのパケットは、TS_recording_rate(TS1)のビットレートでDVR-STDのTBnまたはTBSysのバッファへ入力されなければならない。ここで、TS_recording_rate(TS1)は、Clip1に対応するClipInfo()において定義されるTS_recording_rateの値である。TS1の最後のバイトがバッファへ入力する時刻は、時刻 T_2 であ

る。従って、時刻 T_1 から T_2 までの区間では、ソースパケットのarrival_time_stampは無視される。

【0341】 $N1$ は $TS1$ の最後のビデオパケットに続く $TS1$ のトランスポートパケットのバイト数とすると、時刻 T_1 乃至 T_2 までの時間 $DT1$ は、 $N1$ バイトが $TS_recording_rate(TS1)$ のビットレートで入力終了するために必要な時間であり、次式により算出される。

$$DT1 = T_2 - T_1 = N1 / TS_recording_rate(TS1)$$

時刻 T_1 乃至 T_2 までの間は、 RXn と $RXsys$ の値は共に、 $TS_recording_rate(TS1)$ の値に変化する。このルール以外のバッファリング動作は、 T -STDと同じである。

【0342】 T_2 の時刻において、arrival time clock counterは、 $TS2$ の最初のソースパケットのarrival_time_stampの値にリセットされる。 DVR -STDの $TB1$ 、 TBn または $TBsys$ のバッファへの入力タイミングは、 $TS2$ のソースパケットのarrival_time_stampによって決定される。 RXn と $RXsys$ は共に、 T -STDにおいて定義されている値に変化する。

【0343】付加的なオーディオバッファリングおよびシステムデータバッファリングについて説明するに、オーディオデコーダとシステムデコーダは、時刻 T_1 から T_2 までの区間の入力データを処理することができるように、 T -STDで定義されるバッファ量に加えて付加的なバッファ量（約1秒分のデータ量）が必要である。

【0344】ビデオのプレゼンテーションタイミングについて説明するに、ビデオプレゼンテーションユニットの表示は、接続点を通して、ギャップなしに連続でなければならない。ここで、 $STC1$ は、 $TS1$ のシステムタイムベースの時間軸（図105では $STC1$ と図示されている）とし、 $STC2$ は、 $TS2$ のシステムタイムベースの時間軸（図97では $STC2$ と図示されている。正確には、 $STC2$ は、 $TS2$ の最初のPCRが T -STDに入力した時刻から開始する。）とする。

【0345】 $STC1$ と $STC2$ の間のオフセットは、次のように決定される。 PTS_{end}^1 は、 $TS1$ の最後のビデオプレゼンテーションユニットに対応する $STC1$ 上のPTSであり、 PTS_{start}^2 は、 $TS2$ の最初のビデオプレゼンテーションユニットに対応する $STC2$ 上のPTSであり、 T_{pp} は、 $TS1$ の最後のビデオプレゼンテーションユニットの表示期間とすると、2つのシステムタイムベースの間のオフセット STC_delta は、次式により算出される。

$$STC_delta = PTS_{end}^1 + T_{pp} - PTS_{start}^2$$

【0346】オーディオのプレゼンテーションのタイミングについて説明するに、接続点において、オーディオプレゼンテーションユニットの表示タイミングのオーバーラップがあっても良く、それは0乃至2オーディオフレーム未満である（図105に図示されている"audio overlap"を参照）。どちらのオーディオサンプルを選択するかということと、オーディオプレゼンテーションユニットの表示を接続点の後の補正されたタイムベースに

再同期することは、プレーヤ側により設定されることである。

【0347】 DVR -STDのシステムタイムクロックについて説明するに、時刻 T_5 において、 $TS1$ の最後のオーディオプレゼンテーションユニットが表示される。システムタイムクロックは、時刻 T_2 から T_5 の間にオーバーラップしていても良い。この区間では、 DVR -STDは、システムタイムクロックを古いタイムベースの値（ $STC1$ ）と新しいタイムベースの値（ $STC2$ ）の間で切り替える。 $STC2$ の値は、次式により算出される。

$$STC2 = STC1 - STC_delta$$

【0348】バッファリングの連続性について説明する。 $STC1_{video_end}^1$ は、 $TS1$ の最後のビデオパケットの最後のバイトが DVR -STDの $TB1$ へ到着する時のシステムタイムベース $STC1$ 上の STC の値である。 $STC2_{video_start}^2$ は、 $TS2$ の最初のビデオパケットの最初のバイトが DVR -STDの $TB1$ へ到着する時のシステムタイムベース $STC2$ 上の STC の値である。 $STC2_{video_end}^2$ は、 $STC1_{video_end}^1$ の値をシステムタイムベース $STC2$ 上の値に換算した値である。 $STC2_{video_end}^2$ は、次式により算出される。

$$STC2_{video_end}^2 = STC1_{video_end}^1 - STC_delta$$

【0349】 DVR -STDに従うために、次の2つの条件を満たす事が要求される。まず、 $TS2$ の最初のビデオパケットの $TB1$ への到着タイミングは、次に示す不等式を満たさなければならない。そして、次に示す不等式を満たさなければならない。

$$STC2_{video_start}^2 > STC1_{video_end}^1 + \Delta T_1$$

この不等式が満たされるように、Clip1および、または、Clip2の部分的なストリームを再エンコードおよび、または、再多重化する必要がある場合は、その必要に応じて行われる。

【0350】次に、 $STC1$ と $STC2$ を同じ時間軸上に換算したシステムタイムベースの時間軸上において、 $TS1$ からのビデオパケットの入力とそれに続く $TS2$ からのビデオパケットの入力は、ビデオバッファをオーバーフローおよびアンダーフローさせてはならない。

【0351】このようなシンタックス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0352】なお、本実施の形態は、多重化ストリームとしてMPEG2トランスポートストリームを例にして説明しているが、これに限らず、MPEG2プログラムストリームや米国のDirecTVサービス（商標）で使用されているDSSトランスポートストリームについても適用することが可能である。

【0353】次に、mark_entry()およびrepresentative_picture_entry()のシンタックスが、図81に示される

ような構成である場合における、マーク点で示されるシーンの頭出し再生を行う場合の処理について、図106のフローチャートを参照して、説明する。

【0354】最初にステップS1において、記録再生装置1の制御部23は、記録媒体100から、DVRトランスポートストリームファイルのデータデースであるEP_Map (図70)、STC_Info (図52)、Program_Info (図54)、およびClipMark (図78)を読み出す。

【0355】ステップS2において、制御部23は、ClipMark (図78)のrepresentative_picture_entry (図81)、またはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを作成し、ユーザインターフェイス入出力としての端子24から出力し、GUIのメニュー画面上に表示させる。この場合、ref_thumbnail_indexが有効な値を持つ場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0356】ステップS3において、ユーザが再生開始点のマーク点を指定する。これは、例えば、GUIとして表示されたメニュー画面上の中からユーザがサムネイル画像を選択することで行われる。制御部23は、この選択操作に対応して、指定されたサムネイルに対応づけられているマーク点を取得する。

【0357】ステップS4において、制御部23は、ステップS3で指定されたmark_entry (図81)のmark_time_stampのPTSと、STC_sequence_idを取得する。

【0358】ステップS5において、制御部23は、STC_Info (図52)から、ステップS4で取得したSTC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

【0359】ステップS6において、制御部23は、ステップS5で取得したSTC時間軸が開始するパケット番号と、ステップS4で取得したマーク点のPTSから、マーク点のPTSより時間的に前で、かつ、最も近いエントリーポイント(Iピクチャ)のあるソースパケット番号を取得する。

【0360】ステップS7において、制御部23は、ステップS6で取得したエントリーポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0361】ステップS8において、制御部23は、AVデコーダ27を制御し、ステップS4で取得したマーク点のPTSのピクチャから表示を開始させる。

【0362】以上の動作を、図107乃至109を参照してさらに説明する。

【0363】いま、図107に示されているように、DVRトランスポートストリームファイルは、STC_sequence_id=id0のSTC時間軸を有し、その時間軸が開始するソースパケット番号は、シーン開始点Aのソースパケット番号より小さいものとする。そして、ソースパケット番号BからCまでの間に、CM (コマーシャル)が挿入されて

いるものとする。

【0364】このとき、図70に示されるEP_Mapに対応するEP_Mapには、図108に示されるように、RSPN_EP_startで示されるA、B、Cに対応して、それぞれのPTSが、PTS_EP_startとして、PTS(A)、PTS(B)、PTS(C)として登録される。

【0365】また、図109に示されるように、図78のClipMarkに対応するClipMarkには、図109に示されるように、シーンスタート、CMスタート、およびCMエンドを表すマークタイプ (図79) 0x92、0x94、0x95の値に対応して、mark_entryとrepresentative_picture_entryが記録される。

【0366】mark_entryのMark_Time_stampとしては、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれPTS(a1)、PTS(b0)、PTS(c0)が登録されており、それぞれのSTC_sequence_idは、いずれもid0とされている。

【0367】同様に、Representative_picture_entryのMark_Time_stampとして、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれPTS(a2)、PTS(b0)、PTS(c0)が登録されており、それらはいずれもSTC_sequence_idが、id0とされている。

【0368】PTS(A) < PTS(a1)の場合、ステップS6において、パケット番号Aが取得され、ステップS7において、パケット番号Aから始まるトランスポートストリームが、AVデコーダ27に供給され、ステップS8において、PTS(a1)のピクチャから表示が開始される。

【0369】次に、図110のフローチャートを参照して、mark_entryとrepresentative_picture_entryのシンタクスが、図81に示されるような構成である場合におけるCMスキップ再生の処理について、図110のフローチャートを参照して説明する。

【0370】ステップS21において、制御部23は、EP_map (図70)、STC_Info (図52)、Program_Info (図54)、およびClipMark (図78)を記録媒体100から読み出す。ステップS22において、ユーザは、ユーザインターフェイス入出力としての端子24からCMスキップ再生を指定する。

【0371】ステップS23において、制御部23は、マークタイプ (図79) がCM開始点 (0x94) であるマーク情報のPTSと、CM終了点 (0x95) であるマーク情報のPTS、並びに対応するSTC_sequence_idを取得する (図81)。

【0372】ステップS24において、制御部23は、STC_Info (図52) からCM開始点と終了点の、STC_sequence_idに対応するSTC時間軸が開始するソースパケット番号を取得する。

【0373】ステップS25において、制御部23は、記録媒体100からトランスポートストリームを読み出させ、それをAVデコーダ27に供給し、デコードを開始

させる。

【0374】ステップS26において、制御部23は、現在の表示画像がCM開始点のPTSの画像か否かを調べる。現在の表示画像がCM開始点のPTSの画像でない場合には、ステップS27に進み、制御部23は、画像の表示が継続される。その後、処理はステップS25に戻り、それ以降の処理が繰り返し実行される。

【0375】ステップS26において、現在の表示画像がCM開始点のPTSの画像であると判定された場合、ステップS28に進み、制御部23は、AVデコーダ27を制御し、デコードおよび表示を停止させる。

【0376】次に、ステップS29において、制御部23は、CM終了点のSTC_sequence_idに対応するSTC時間軸が開始するパケット番号を取得し、そのパケット番号と、ステップS23の処理で取得したCM終了点のPTSとから、その点のPTSより時間的に前で、かつ、最も近いエントリーポイントのあるソースパケット番号を取得する。

【0377】ステップS30において、制御部23は、ステップS29の処理で取得したエントリーポイントのあるソースパケット番号から、トランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0378】ステップS31において、制御部23は、AVデコーダ27を制御し、CM終了点のPTSのピクチャから表示を再開させる。

【0379】図107乃至図109を参照して、以上の動作をさらに説明すると、CM開始点とCM終了点は、この例の場合、STC_sequence_id=id0という共通のSTC時間軸上に存在し、そのSTC時間軸が開始するソースパケット番号は、シーンの開始点のソースパケット番号Aより小さいものとされている。

【0380】トランスポートストリームがデコードされ、ステップS26で、表示時刻がPTS(b0)になったと判定された場合（CM開始点であると判定された場合）、AVデコーダ27により表示が停止される。そして、PTS(C)<PTS(c0)の場合、ステップS30でパケット番号Cのデータから始まるストリームからデコードが再開され、ステップS31において、PTS(c0)のピクチャから表示が再開される。

【0381】なお、この方法は、CMスキップ再生に限らず、一般的にClipMarkで指定される2点間のシーンをスキップして再生する場合にも、適用可能である。

【0382】次に、mark_entryとrepresentative_picture_entryが、図82に示すシンタックス構造である場合における、マーク点で示されるCMの頭出し再生処理について、図112のフローチャートを参照して説明する。

【0383】ステップS41において、制御部23は、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、およびClipMark（図78）の情報を取得する。

【0384】次にステップS42において、制御部23は、ステップS41で読み出したClipMark（図78）に含まれるrepresentative_picture_entry（図82）またはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面上に表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0385】ステップS43において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、ステップS42の処理で表示されたメニュー画面上の中から、ユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を指定することで行われる。

【0386】ステップS44において、制御部23は、ステップS43の処理で指定されたマーク点のRSPN_ref_EP_startとoffset_num_pictures（図82）を取得する。

【0387】ステップS45において、制御部23は、ステップS44で取得したRSPN_ref_EP_startに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0388】ステップS46において、制御部23は、AVデコーダ27を制御し、RSPN_ref_EP_startで参照されるピクチャから（表示はしないで）、表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_picturesになったとき、そのピクチャから表示を開始させる。

【0389】以上の処理を、図113乃至図115を参照して、さらに説明する。この例においては、DVRトランスポートストリームファイルは、ソースパケット番号Aからシーンが開始しており、ソースパケット番号BからソースパケットCまでCMが挿入されている。このため、図114に示されるように、EP_mapには、RSPN_EP_startとしてのA、B、Cに対応して、PTS_EP_startとして、PTS(A)、PTS(B)、PTS(C)が登録されている。

【0390】また、図115に示されるように、シーンスタート、CMスタート、およびCMエンドのマークタイプに対応して、mark_entryとrepresentative_picture_entryが登録されている。mark_entryには、シーンスタート、CMスタート、およびCMエンドに対応して、RSPN_ref_EP_startとして、それぞれA、B、Cが登録され、offset_num_picturesとして、M1、N1、N2が登録されている。同様に、representative_picture_entryには、RSPN_ref_EP_startとして、シーンスタート、CMスタート、およびCMエンドに対応して、それぞれA、B、Cが登録され、offset_num_picturesとして、M2、N1、N2がそれぞれ登録されている。

【0391】シーンスタートに当たるピクチャから頭出しして再生が指令された場合、パケット番号Aのデータか

ら始まるストリームからデコードが開始され、PTS(A)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップをしていき、offset_num_picturesが、M1の値になったとき、そのピクチャから表示が開始される。

【0392】さらに、mark_entryとrepresentative_picture_entryのシンタクスが、図82に示される構成である場合におけるCMスキップ再生の処理について、図116のフローチャートを参照して説明する。

【0393】ステップS61において、制御部23は、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、およびClipMark（図78）の情報を取得する。

【0394】ステップS62において、ユーザがCMスキップ再生を指令すると、ステップS63において、制御部23は、マークタイプ（図79）がCM開始点とCM終了点である各点のマーク情報として、RSPN_ref_EP_STARTとoffset_num_pictures（図82）を取得する。そして、CM開始点のデータは、RSPN_ref_EP_start(1)、offset_num_pictures(1)とされ、CM終了点のデータは、RSPN_ref_EP_start(2)、offset_num_pictures(2)とされる。

【0395】ステップS64において、制御部23は、RSPN_ref_EP_start(1)、RSPN_ref_EP_start(2)に対応するPTSをEP_map（図70）から取得する。

【0396】ステップS65において、制御部23は、トランスポートストリームを記録媒体100から読み出させ、AVデコーダ27に供給させる。

【0397】ステップS66において、制御部23は、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであるか否かを判定し、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャでない場合には、ステップS67に進み、ピクチャをそのまま継続的に表示させる。その後、処理はステップS65に戻り、それ以降の処理が繰り返し実行される。

【0398】ステップS66において、現在の表示画像がRSPN_ref_EP_start(1)に対応するPTSのピクチャであると判定された場合、ステップS68に進み、制御部23は、AVデコーダ27を制御し、RSPN_ref_EP_start(1)に対応するPTSのピクチャから表示するピクチャをカウントアップしていき、カウント値がoffset_num_pictures(1)になったとき、表示を停止させる。

【0399】ステップS69において、制御部23は、RSPN_ref_EP_start(2)のソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0400】ステップS70において、制御部23は、AVデコーダ27を制御し、RSPN_ref_EP_start(2)に対応するPTSのピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、カウント値がoffset_num_pictures(2)になったとき、そのピクチャから表

示を開始させる。

【0401】以上の動作を、図113乃至図115を参照してさらに説明すると、まず、EP_map（図114）をもとに、パケット番号B、Cに対応する時刻PTS(B)、PTS(C)が得られる。そして、Clip AV streamがデコードされていき、表示時刻がPTS(B)になったとき、PTS(B)のピクチャから表示ピクチャがカウントアップされ、その値がN1（図115）になったとき、表示が停止される。

【0402】さらに、パケット番号Cのデータから始まるストリームからデコードが再開され、PTS(C)のピクチャから（表示をしないで）表示すべきピクチャをカウントアップしていき、その値がN2（図115）になったとき、そのピクチャから表示が再開される。

【0403】以上の処理は、CMスキップ再生に限らず、ClipMarkで指定された2点間のシーンをスキップさせて再生する場合にも、適用可能である。

【0404】次に、mark_entryとrepresentative_picture_entryのシンタクスが、図84に示すような構成である場合における、マーク点で示されるシーンの頭出し再生処理について、図118のフローチャートを参照して説明する。

【0405】ステップS81において、EP_map（図70）、STC_Info（図52）、Program_Info（図54）、並びにClipMark（図78）の情報が取得される。

【0406】ステップS82において、制御部23は、ClipMark（図78）のrepresentative_picture_entryまたはref_thumbnail_indexで参照されるピクチャからサムネイルのリストを生成し、GUIのメニュー画面として表示させる。ref_thumbnail_indexが有効な値を有する場合、representative_picture_entryよりref_thumbnail_indexが優先される。

【0407】ステップS83において、ユーザは再生開始点のマーク点を指定する。この指定は、例えば、メニュー画面上の中からユーザがサムネイル画像を選択し、そのサムネイルに対応づけられているマーク点を指定することで行われる。

【0408】ステップS84において、制御部23は、ユーザから指定されたmark_entryのRSPN_mark_point（図84）を取得する。

【0409】ステップS85において、制御部23は、マーク点のRSPN_mark_pointより前にあり、かつ、最も近いエン트리ポイントのソースパケット番号を、EP_map（図70）から取得する。

【0410】ステップS86において、制御部23は、ステップS85で取得したエン트리ポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0411】ステップS87において、制御部23は、AVデコーダ27を制御し、RSPN_mark_pointで参照されるピクチャから表示を開始させる。

【0412】以上の処理を、図119乃至図121を参照してさらに説明する。この例においては、DVRトランスポートストリームファイルが、ソースパケットAでシーンスタートし、ソースパケット番号BからCまでCMが挿入されている。このため、図120のEP_mapには、RSPN_EP_startとしてのA、B、Cに対応して、PTS_EP_startがそれぞれPTS(A)、PTS(B)、PTS(C)として登録されている。また、図121に示されるClipMarkに、シーンスタート、CMスタート、およびCMエンドに対応して、mark entryのRSPN_mark_pointとして、a1、b1、c1が、また、representative_picture_entryのRSPN_mark_pointとして、a2、b1、c1が、それぞれ登録されている。

【0413】シーンスタートにあたるピクチャから頭出して再生する場合、パケット番号A<a1とすると、パケット番号Aのデータから始まるストリームからデコードが開始され、ソースパケット番号a1に対応するピクチャから表示が開始される。

【0414】次に、mark_entryとrepresentative_picture_entryのシンタックスが、図84に示されるような構成である場合におけるCMスキップ再生の処理について、図122と図123のフローチャートを参照して説明する。

【0415】ステップS101において、制御部23は、EP_map (図70)、STC_Info (図52)、Program_Info (図54)、並びにClipMark (図70)の情報を取得する。

【0416】ステップS102において、ユーザは、CMスキップ再生を指定する。

【0417】ステップS103において、制御部23は、マークタイプ (図79)がCM開始点とCM終了点である各点のマーク情報のRSPN_mark_point (図84)を取得する。そして、制御部23は、CM開始点のデータをRSPN_mark_point (1)とし、CM終了点のデータをRSPN_mark_point (2)とする。

【0418】ステップS104において、制御部23は、記録媒体100からトランスポートストリームを読み出させ、AVデコーダ27に出力し、デコードさせる。

【0419】ステップS105において、制御部23は、現在の表示画像がRSPN_mark_point (1)に対応するピクチャであるか否かを判定し、現在の表示画像がRSPN_mark_point (1)に対応するピクチャでない場合には、ステップS106に進み、そのままピクチャを継続的に表示させる。その後、処理はステップS104に戻り、それ以降の処理が繰り返し実行される。

【0420】ステップS105において、現在の表示画像がRSPN_mark_point (1)に対応するピクチャであると判定された場合、ステップS107に進み、制御部23はAVデコーダ27を制御し、デコードおよび表示を停止させる。

【0421】次に、ステップS108において、RSPN_mark_point (2)より前にあり、かつ、最も近いエンタリーポイントのあるソースパケット番号がEP_map (図70)から取得される。

【0422】ステップS109において、制御部23は、ステップS108で取得したエンタリーポイントに対応するソースパケット番号からトランスポートストリームのデータを読み出し、AVデコーダ27に供給させる。

【0423】ステップS110において、制御部23は、AVデコーダ27を制御し、RSPN_mark_point (2)で参照されるピクチャから表示を再開させる。

【0424】以上の処理を図119乃至図121の例でさらに説明すると、Clip AV streamをデコードして行き、ソースパケット番号b1 (図121)に対応する表示ピクチャになったとき、表示が停止される。そして、ソースパケット番号C<ソースパケット番号c1とすると、パケット番号Cのデータから始まるストリームからデコードが再開され、ソースパケット番号c1に対応するピクチャになったとき、そのピクチャから表示が再開される。

【0425】以上のようにして、図124に示されるように、PlayList上で、タイムスタンプにより所定の位置を指定し、このタイムスタンプを各ClipのClip Informationにおいて、データアドレスに変換し、Clip AV streamの所定の位置にアクセスすることができる。

【0426】より具体的には、図125に示されるように、PlayList上において、PlayListMarkとしてブックマークやリジューム点を、ユーザが時間軸上のタイムスタンプとして指定すると、そのPlayListは再生するとき、そのPlayListが参照しているClipのClipMarkを使用して、Clip AV streamのシーン開始点やシーン終了点にアクセスすることができる。

【0427】なお、ClipMarkのシンタックスは、図78の例に替えて、図126に示すようにすることもできる。

【0428】この例においては、RSPN_markが、図78のreserved_for_MakerID、mark_entry ()、およびrepresentative_picture_entry ()に替えて挿入されている。このRSPN_markの32ビットのフィールドは、AVストリームファイル上で、そのマークが参照するアクセスユニットの第1バイト目を含むソースパケットの相対アドレスを示す。RSPN_markは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットからClip Information fileにおいて定義され、offset_SPNの値を初期値としてカウントされる。

【0429】その他の構成は、図78における場合と同様である。

【0430】ClipMarkのシンタックスは、さらに図127に示すように構成することもできる。この例において

は、図126におけるRSPN_markの代わりに、RSPN_ref_EP_startとoffset_num_picturesが挿入されている。これらは、図82に示した場合と同様のものである。

【0431】次に、図128は、ClipInfo()のシンタクスの別例を示す。

【0432】Clip_service_typeは、AVストリームファイルのタイプを示す。例えば、Clip_service_typeは、ビデオレコーディングやオーディオレコーディングなどのタイプを示す。また、例えば、Clip_service_typeは、デジタルTV放送のプログラムが示すサービスタイプと同じ意味を持たせても良い。例えば、日本のデジタルBS放送の場合、サービスタイプは、テレビサービス、音声サービスおよびデータ放送サービスの3種類を持つ。AVストリームが含むプログラムのサービスタイプを代表する値をClip_service_typeにセットする。

【0433】transcode_mode_flagは、デジタル放送から受信されたMPEG2 トランスポートストリームの記録方法を示すフラグである。このフラグが1にセットされている場合、Clipに対応するAVストリームファイル中の少なくとも1つのエレメンタリーストリームは再符号化されて記録されたことを示す。このフラグが1にセットされている場合、Clipに対応するAVストリームファイル中のすべてのエレメンタリーストリームはデジタル放送から受信されたままの内容で何も変更されないで記録されたことを示す。

【0434】その他のシンタクスフィールドは、図46で説明した同名のフィールドと同じ意味を持つ。

【0435】次に、図129を参照して、ProgramInfo()の別例について説明する。

【0436】AVストリームファイルの中で本フォーマットが規定するところのプログラム内容が一定であるソースパケット列を、program-sequenceと呼ぶ。

【0437】AVストリームファイルの中で、新しいprogram-sequenceが開始するアドレスをProgramInfo()にストアする。このアドレスは、SPN_program_sequence_startにより示される。

【0438】AVストリームファイルの中にある最後のprogram-sequence以外のprogram-sequenceは、そのSPN_program_sequence_startで指されるソースパケットから開始し、その次のSPN_program_sequence_startで指されるソースパケットの直前のソースパケットで終了する。最後のprogram-sequenceは、そのSPN_program_sequence_startで指されるソースパケットから開始し、AVストリームファイルの最後のソースパケットで終了する。

【0439】program-sequenceは、STC-sequenceの境界をまたいで良い。

【0440】lengthは、このlengthフィールドの直後のバイトからProgramInfo()の最後のバイトまでのバイト数を示す。

【0441】num_of_program_sequencesは、AVストリー

ムファイルの中にあるprogram-sequenceの数を示す。

【0442】SPN_program_sequence_startは、AVストリームファイル上でprogram-sequenceが開始するアドレスを示す。SPN_program_sequence_startは、ソースパケット番号を単位とする大きさであり、AVストリームファイルの最初のソースパケットから、ゼロを初期値としてカウントされる。

【0443】ProgramInfo()の中でエントリーされるSPN_program_sequence_startの値は、昇順に並んでいる。

【0444】SPN_program_sequence_startは、そのprogram-sequenceに対する最初のPMTを持つソースパケットを指していることを前提とする。SPN_program_sequence_startは、記録機(図1の記録再生装置1)がトランスポートストリーム中のPSI/SIを解析することによって作られる。記録機がPSI/SIを解析し、その変化を検出するまでの遅延時間が必要なために、SPN_program_sequence_startは、実際のPSI/SIの変化点から所定の時間以内にあるソースパケットを指しても良い。

【0445】program_map_PIDは、そのprogram-sequenceに適用できるPMT(program map table)を持つトランスポートパケットのPIDの値である。

【0446】num_of_streams_in_psは、そのprogram-sequenceの中で定義されるエレメンタリーストリームの数を示す。

【0447】num_of_groupsは、そのprogram-sequenceの中で定義されるエレメンタリーストリームのグループの数を示す。num_of_groupsは、1以上の値である。

【0448】トランスポートストリームのPSI/SIがエレメンタリーストリームのグループ情報を持つ場合、num_of_groupsは、1以上の値をとることを想定している。それぞれのグループは、マルチ・ビュー・プログラム中の1つのビューを構成する。

【0449】stream_PIDは、そのprogram-sequenceのprogram_map_PIDが参照するところのPMTの中で定義されているエレメンタリーストリームに対するPIDの値を示す。

【0450】StreamCodingInfo()は、上記stream_PIDで指されるエレメンタリーストリームの情報を示す。詳細は後述する。

【0451】num_of_streams_in_groupは、エレメンタリーストリームのグループが持つエレメンタリーストリームの数を示す。

【0452】stream_indexは、上記エレメンタリーストリームのグループが持つエレメンタリーストリームに対応するところの、シンタクス中のstream_indexのfor-loopで記述される順番で定義されるstream_indexの値を示す。

【0453】図30は、StreamCodingInfo()のシンタクスを示す。

【0454】lengthは、このlengthフィールドの直後の

バイトからStreamCodingInfo()の最後のバイトまでのバイト数を示す。

【0455】stream_coding_typeは、このStreamCodingInfo()に対応するstream_PIDで指されるエレメンタリーストリームの符号化タイプを示す。値の意味は、図131に示される。

【0456】video_formatは、このStreamCodingInfo()に対応するstream_PIDで指されるビデオストリームのビデオフォーマットを示す。

【0457】値の意味は、図56と同じである。

【0458】frame_rateは、このStreamCodingInfo()に対応するstream_PIDで指されるビデオストリームのフレームレートを示す。

【0459】値の意味は、図57と同じである。

【0460】display_aspect_ratioは、このStreamCodingInfo()に対応するstream_PIDで指されるビデオストリームのディスプレイ・アスペクト・レシオを示す。

【0461】値の意味は、図58と同じである。

【0462】cc_flagは、このStreamCodingInfo()に対応するstream_PIDで指されるビデオストリームの中でクロズド・キャプション(closed caption data)信号が符号化されているか否かを示すフラグである。

【0463】original_video_format_flagは、このStreamCodingInfo()の中にoriginal_video_formatとoriginal_display_aspect_ratioが存在するか否かを示すフラグである。

【0464】original_video_formatは、このStreamCodingInfo()に対応するstream_PIDで指されるビデオストリームが符号化される前のオリジナルのビデオフォーマットである。値の意味は、上記のvideo_format(図56)と同じである。

【0465】original_display_aspect_ratioは、このStreamCodingInfo()に対応するstream_PIDで指されるビデオストリームが符号化される前のオリジナルのディスプレイ・アスペクト・レシオである。値の意味は、上記のdisplay_aspect_ratio(図58)と同じである。

【0466】ビデオストリームと共にマルチメディアデータストリーム(BMLストリーム、字幕など)が多重化されているトランスポートストリームをトランス・コーディングする場合において、ビデオストリームは再エンコードされることによって、そのビデオフォーマットが変化する(例えば、1080iから480iへ変化する)、一方、マルチメディアデータストリームはオリジナルのストリームのままの内容を保つ場合を考える。この時、新しいビデオストリームとマルチメディアデータストリームの間に情報のミスマッチが生じる場合がある。例えば、マルチメディアデータストリームの表示に関するパラメータは、オリジナルのビデオストリームのビデオフォーマットを想定して決められているにもかかわらず、ビデオストリームの再エンコードによって、そのビデオ

フォーマットが変化した場合である。このような場合、original_video_formatとoriginal_display_aspect_ratioに、オリジナルのビデオストリームに関する情報を保存する。再生機(図1の記録再生装置1)は、上記の新しいビデオストリームとマルチメディアデータストリームから次のようにして、表示画像をつくる。

【0467】・ビデオストリームは、original_video_formatとoriginal_display_aspect_ratioで示されるビデオフォーマットにアップ・サンプリングされる。・そのアップ・サンプリングされた画像とマルチメディアデータストリームが合成されて、正しい表示画像をつくる。

【0468】audio_presentation_typeは、このStreamCodingInfo()に対応するstream_PIDで指されるオーディオストリームのプレゼンテーション・タイプを示す。

【0469】値の意味は、図61のaudio_component_typeと同じである。

【0470】sampling_frequencyは、このStreamCodingInfo()に対応するstream_PIDで指されるオーディオストリームのサンプリング周波数を示す。

【0471】値の意味は、図62と同じである。

【0472】次に、EP_mapの別例を示す。このEP_mapは、図70と図72で説明したPTS_EP_startとRSPN_EP_startのデータをそのままEP_mapにストアするのではなく、データ量の削減のために圧縮符号化して(図1の制御部23が実行する)EP_mapにストアする。

【0473】EP_mapは、少なくとも1つ以上のEP_map_for_one_stream_PID()と呼ばれるサブテーブルから成る。サブテーブルEP_map_for_one_stream_PID()は、同じPIDの値を持つトランスポートパケットによって伝送される1つのエレメンタリーストリームに対して作られる。

【0474】EP_mapは、すべてのEP_map_for_one_stream_PID()のための次の情報を持つ。

【0475】(1) stream_PID: そのEP_map_for_one_stream_PID()によって指されるエレメンタリーストリームを伝送するところのトランスポートパケットのPIDを特定する。

(2) EP_stream_type: そのEP_map_for_one_stream_PID()のエントリーポイントのタイプを示す。

(3) EP_map_for_one_stream_PID()のエントリー。それぞれのエントリーは次の情報を持つ：

・PTS_EP_start: エントリーポイントにおけるアクセスユニットのPTSの値を示す。

・RSPN_EP_start: AVストリームの中で、そのエントリーポイントのPTS_EP_startによって参照されるところのアクセスユニットの開始するアドレスを示す。

【0476】EP_mapが1つ以上のエレメンタリーストリームを指す場合、EP_mapは1つ以上のEP_map_for_one_stream_PID()サブテーブルを持っても良い。図67は、EP_mapがClip中の3個のビデオストリームを指している

場合のEP_mapの例を示している。

【0477】1つのEP_map_for_one_stream_PID()は、STCの不連続点に関係なく1つのテーブルに作られる。

【0478】RSPN_EP_startの値とSTC_Info()において定義されるRSPN_STC_startの値を比較する事により、EP_mapの中にあるSTC_sequenceの境界が分かる(図68参照)。

【0479】EP_mapは、同じPIDで伝送されるストリームの連続した範囲に対して、1つのEP_map_for_one_stream_PIDを持つ。

【0480】図69の場合、program#1とprogram#3は、同じビデオPIDを持つが、データ範囲が連続していないので、それぞれのプログラム毎にEP_map_for_one_stream_PIDを持たねばならない。

【0481】EP_map_for_one_stream_PID()のテーブルのデータサイズを削減し、かつデータサーチのパフォーマンスを改善するために、EP_map_for_one_stream_PID()は2つのサブテーブル、すなわち、EP_coarseとEP_fineに分けられる(図132参照)。

【0482】EP_fineエントリーは、PTS_EP_start および RSPN_EP_startのLSB(Least Significant Bit)側のビット情報を持つ(図133と図134を参照)。EP_coarseエントリーは、PTS_EP_start および RSPN_EP_startのMSB(Most Significant Bit)側のビット情報とそれに対応するEP_fineのテーブル中のエントリー番号(同じPTS_EP_startから取り出したLSB側のビットを持つEP_fineテーブルの中のエントリー)を持つ。

【0483】EP_coarseサブテーブルの中のエントリー数は、EP_fineサブテーブルのエントリー数に比べてかなり小さい。次の場合に、EP_coarseのエントリーが作られる：

- ・最初のPTS_EP_fineのエントリー
- ・PTS_EP_fineの値がラップ・アラウンドした後の最初のPTS_EP_fineのエントリー
- ・RSPN_EP_fineの値がラップ・アラウンドした後の最初のRSPN_EP_fineのエントリー(図135参照)。

【0484】EP_mapを使用する場合のAVストリームへのランダムアクセスの例について説明する。

【0485】あるPlaylistのグローバルな時間軸上で30分後から、その再生を開始したいとする。

- ・最初にPlaylistの中のグローバルな時間軸上で30分後に対応するところ時刻を含むPlayItemのSTC-sequence-idを調べる。

・前記STC-sequenceのローカルな時間軸上で30分後に対応するところのPTSの値を導く。

・前記STC-sequenceのRSPN_STC_startをSTC_Infoから導く。

・EP_coarse サブテーブルの中で、RSPN_EP_coarseが前記RSPN_STC_start以上であるところのエントリーから、データサーチが開始される。EP_coarse サブテーブルの

中で、前記必要なPTSに最も近く、かつその値より時間的に前の値を持つPTS_EP_coarseのエントリーを見つける。

・EP_fineサブテーブルの中で、前記見つけたPTS_EP_coarseに対応するところのEP_fineのエントリー番号から、データサーチが開始される。前記必要なPTSに最も近く、かつその値より時間的に前の値を持つPTS_EP_fineを持つエントリーを見つける。

・前記必要なPTSのアクセスユニットの読み出し開始のためのRSPN_EP_startは、前記見つけたPTS_EP_coarseに対応するRSPN_EP_coarseのLSB 18ビットを前記見つけたPTS_EP_fineに対応するRSPN_EP_fineのビットで置きかえることにより決定される。

【0486】上記説明したEP_mapのシンタックスを図136に示す。

【0487】number_of_stream_PID_entriesは、EP_mapの中でのEP_map_for_one_stream_PIDのエントリー数を示す。

【0488】stream_PID[k]は、EP_mapの中でk番目にエントリーされるEP_map_for_one_stream_PIDによって参照されるエレメンタリーストリームを伝送するところのトランスポートバケットのPIDの値を示す。

【0489】EP_stream_type[k] は、前記EP_map_for_one_stream_PIDのによって参照されるエレメンタリーストリームのタイプを示す。値の意味を図137の表に示す。

【0490】EP_stream_type[k]が0または1の場合、そのエレメンタリーストリームはビデオストリームである。Video type 1 と video type 2 の意味についてはEP_video_type(図139)の説明のところで後述する。

【0491】EP_stream_type[k]が2の場合、そのエレメンタリーストリームはオーディオストリームである。

【0492】num_EP_coarse_entries[k] は、前記EP_map_for_one_stream_PIDの中にあるEP-coarseエントリーの数を示す。

【0493】num_EP_fine_entries[k] は、前記EP_map_for_one_stream_PIDの中にあるEP-fineエントリーの数を示す。

【0494】EP_map_for_one_stream_PID_start_addresses[k]は、EP_map()の中で前記EP_map_for_one_stream_PIDが始まる相対バイト位置を示す。この値は、EP_map()の第1バイト目からのバイト数で示される。

【0495】EP_map_for_one_stream_PIDのシンタックスを図138に示す。これのセマンティクスを説明するために、EP_map_for_one_stream_PIDにストアされるデータの元となるところのPTS_EP_start と RSPN_EP_startの意味について説明する。

【0496】PTS_EP_startとそれに関連付けられたRSPN_EP_startは、AVストリーム上のエントリーポイントを

指す。そして、PTS_EP_fineとそれに関連付けられたPTS_EP_coarseは、同一のPTS_EP_startから導かれる。また、RSPN_EP_fineとそれに関連付けられたRSPN_EP_coarseは同じRSPN_EP_startから導かれる。PTS_EP_start と RSPN_EP_startは次のように定義される。

【0497】PTS_EP_startは、33-bit長の符号なし整数である。PTS_EP_startの定義は、EP_map_for_one_stream_PIDに対するEP_stream_typeの値により異なる。

【0498】EP_stream_typeがゼロであるとき('video type 1')、PTS_EP_startは、AVストリームの中でシーケンスヘッダから開始するビデオアクセスユニットの33ビット長のPTSを示す。

【0499】EP_stream_typeが2であるとき('audio')、PTS_EP_startは、AVストリームの中でシーケンスヘッダから開始するビデオアクセスユニットの33ビット長のPTSを示す。

【0500】EP_stream_typeが1であるとき('video type 2')、PTS_EP_startは、それに関連づけられたEP_video_typeの値に従って、図139に定義されるところのビデオアクセスユニットの33ビット長のPTSを示す。

【0501】RSPN_EP_startは、32ビットの符号なし整数である。RSPN_EP_startの定義はEP_map_for_one_stream_PIDに対するEP_stream_typeの値により異なる。

【0502】EP_stream_typeがゼロであるとき('video type 1')、このフィールドはPTS_EP_start に関連付けられたビデオアクセスユニットの第1バイト目を含むソースパケットのAVストリームの中でのアドレスを示す。

【0503】EP_stream_typeが2であるとき('audio')、このフィールドはPTS_EP_start に関連付けられたオーディオアクセスユニットの第1バイト目を含むソースパケットのAVストリームの中でのアドレスを示す。

【0504】EP_stream_typeが1であるとき('video type 2')、RSPN_EP_startの意味は、それに関連づけられたEP_video_typeの値に従って、図139に定義されるものである。

【0505】RSPN_EP_startは、ソースパケット番号の単位で表され、それはAVストリームファイルの中の最初のソースパケットからゼロを初期値としてカウントされる。

【0506】EP_map_for_one_stream_PIDのセマンティクスを説明する。

【0507】EP_fine_table_start_addressは、最初のEP_video_type[EP_fine_id]の第1バイト目の開始アドレスを示し、それはEP_map_for_one_stream_PID()の第1バイト目からの相対バイト数で表される。相対バイト数はゼロから開始する。

【0508】ref_to_EP_fine_idは、このフィールドに続くPTS_EP_coarseに関連づけられるところのPTS_EP_fi

neを持つEP_fineエントリーの番号を示す。PTS_EP_fineとそれに関連付けられるPTS_EP_coarseは、同じPTS_EP_startから導かれる。

【0509】ref_to_EP_fine_idは、EP_fine_idのfor-loopの中で記述される順番で定義されるところのEP_fine_idの値により与えられる。

【0510】PTS_EP_coarseとPTS_EP_fine、そしてRSPN_EP_coarseとRSPN_EP_fineは、次のように導かれる。

【0511】EP_fineサブテーブルに、Nf個のエントリーがあり、これらのエントリーはそれらに関連するRSPN_EP_startの値の昇順に並んでいるとする。

【0512】それぞれのPTS_EP_fineエントリーは、そのPTS_EP_startから次のように導かれる。

【0513】 $PTS_EP_fine[EP_fine_id] = (PTS_EP_start[EP_fine_id] > ; > ; 9) \% 2^{12}$

【0514】PTS_EP_coarse とそのPTS_EP_fineの関係は次のとおりである。

【0515】 $PTS_EP_coarse[i] = (PTS_EP_start[ref_to_EP_fine_id[i]] > ; > ; 19) \% 2^{14}$

$TS_EP_fine[ref_to_EP_fine_id[i]] = (PTS_EP_start[ref_to_EP_fine_id[i]] > ; > ; 9) \% 2^{12}$

【0516】それぞれのRSPN_EP_fineエントリーは、そのRSPN_EP_startから次のように導かれる。

【0517】 $RSPN_EP_fine[EP_fine_id] = RSPN_EP_start[EP_fine_id] \% 2^{18}$

【0518】RSPN_EP_coarse とその RSPN_EP_fineの関係は次のとおりである。

【0519】 $RSPN_EP_coarse[i] = RSPN_EP_start[ref_to_EP_fine_id[i]]$

$RSPN_EP_fine[ref_to_EP_fine_id[i]] = RSPN_EP_start[ref_to_EP_fine_id[i]] \% 2^{18}$

【0520】EP_video_type[EP_fine_id]は、EP_stream_typeが1('video type 2')である場合、PTS_EP_fine[EP_fine_id]およびRSPN_EP_fine[EP_fine_id]が指すところのビデオアクセスユニットのタイプを示す。値の意味は、図139に示すとおりである。

【0521】EP_map_for_one_stream_PID()は次の制限を満たす。

【0522】EP_map_for_one_stream_PID()の中でRSPN_EP_coarseのエントリーは、RSPN_EP_coarseの値の昇順で並んでいなければならない。

EP_map_for_one_stream_PID()の中でRSPN_EP_fineのエントリーは、それに関連するRSPN_EP_startの値の昇順で並んでいなければならない。

・次の場合に、PTS_EP_coarseとRSPN_EP_coarseのエントリーが作られる：

・最初のPTS_EP_fineのエントリー

・PTS_EP_fineの値がラップ・アラウンドした後の最初のPTS_EP_fineのエントリー

・RSPN_EP_fineの値がラップ・アラウンドした後の最

初のRSPN_EP_fineのエントリー。

【0523】次に、図140はClip AVストリームファイルとそれに関連するClip Informationファイルの記録動作のフローチャートを示す。図1の記録再生装置を参照して説明する。

【0524】ステップS201で、制御部23は、端子11および12から入力されるAV入力をエンコードして得たトランスポートストリーム、または端子13のデジタルインタフェースから入力されるトランスポートストリームをファイル化して、Clip AVストリームファイルを作成して記録する。

【0525】ステップS202で、制御部23は上記AVストリームファイルについてのClipInfoを作成する。

【0526】ステップS203で、制御部23は上記AVストリームファイルについてのSTC_Infoを作成する。

【0527】ステップS204で、制御部23は上記AVストリームファイルについてのProgram_Infoを作成する。

【0528】ステップS205で、制御部23は上記AVストリームファイルについてのCPI (EP-mapまたはTU-map)を作成する。

【0529】ステップS206で、制御部23は上記AVストリームファイルについてのClipMarkを作成する。

【0530】ステップS207で、制御部23は上記ClipInfo、STC_Info、ProgramInfo、CPI、およびClipMarkがストアされたClip Informationファイルを記録する。

【0531】なお、ここでは各処理を時系列に説明したが、ステップS201からステップS206は実際には同時に動作するものである。

【0532】次に、STC_Infoの作成の動作例を図14のフローチャートを用いて説明する。この処理は図1の多重化ストリーム解析部18で行われる。

【0533】ステップS221において、ストリーム解析部18はPCRパケット受信されたかどうかを調べる。ステップS221において、Noの場合はステップS221へ戻り、Yesの場合はステップS222へ進む。

【0534】ステップS222において、STCの不連続が検出されたかを調べる。Noの場合は、ステップS221へ戻る。Yesの場合は、ステップS223へ進む。なお、記録開始後、最初に受信されたPCRパケットの場合は、必ずステップS223へ進む。

【0535】ステップS223において、新しいSTCの最初のPCRを送信するトランスポートパケットの番号(アドレス)を取得する。

【0536】ステップS224において、STC_Infoを作成する。

【0537】ステップS225において、最後のトランスポートパケットが入力終了したかどうかを調べる。Noの場合は、ステップS221へ戻り、Yesの場合は処理を終了する。

【0538】Program_Infoの作成の動作例を図142のフローチャートを用いて説明する。この処理は図1の多重化ストリーム解析部18で行われる。

【0539】ステップS241において、ストリーム解析部18はPSI/SIを含むトランスポートパケットが受信されたかどうかを調べる。ここで、PSI/SIのトランスポートパケットは、具体的には、PAT、PMT、SITのパケットである。SITは、DVB規格で規定されているパースナルトランスポートストリームのサービス情報が記述されているトランスポートパケットである。ステップ241において、Noの場合はステップS241へ戻り、Yesの場合はステップS242へ進む。

【0540】ステップS242において、PSI/SIの内容が変わったかを調べる。すなわち、PAT、PMT、SITのそれぞれの内容が、以前に受信したそれぞれの内容と比べて変化したかどうかを調べる。内容が変化していない場合は、ステップS241へ戻る。内容が変化した場合は、ステップS243へ進む。なお、記録開始後、最初に受信されたPSI/SIの場合は、必ずステップS243へ進む。

【0541】ステップS243において、新しいPSI/SIを送信するトランスポートパケットの番号(アドレス)とその内容を取得する。

【0542】ステップS244において、Program-sequeneの情報を作成する。

【0543】ステップS245において、最後のトランスポートパケットが入力終了したかどうかを調べる。Noの場合は、ステップS241へ戻り、Yesの場合は処理を終了する。

【0544】次にEP_mapの作成の動作例を図143のフローチャートを用いて説明する。この処理は図1の多重化ストリーム解析部18で行われる。

【0545】ステップS261でストリーム解析部18は、記録するAVプログラムのビデオのPIDをセットする。トランスポートストリームの中に複数のビデオが含まれている場合は、それぞれのビデオPIDをセットする。

【0546】ステップS262でストリーム解析部18は、ビデオのトランスポートパケットを受信する。

【0547】ステップS263でストリーム解析部は、トランスポートパケットのペイロード(パケットヘッダーに続くデータ部)がPESパケットの第1バイト目から開始しているかを調べる(PESパケットは、MPEG2で規定されているパケットであり、エレメンタリーストリームをパケット化するものである)。これは、トランスポートパケットヘッダにある"payload_unit_start_indicator"の値を調べることによりわかり、この値が1である場合、トランスポートパケットのペイロードがPESパケットの第1バイト目から開始する。ステップS263でNoの場合は、ステップS262へ戻り、Yesの場合は、ス

テップS 264へ進む。

【0548】ステップS 264でストリーム解析部は、PESパケットのペイロードが、MPEGビデオのsequence_header_code(32ビット長で"0x000001B3"の符号)の第1バイト目から開始しているかを調べる。ステップS 264でNoの場合は、ステップS 262へ戻り、Yesの場合は、ステップS 265へ進む。

【0549】ステップS 265へ進んだ場合、現在のトランスポートパケットをエン트리ポイントとする。

【0550】ステップS 266でストリーム解析部は、上記パケットのパケット番号と上記sequence_header_code から開始するIピクチャのPTSとそのエン트리ポイントが属するビデオのPIDを取得し、制御部23へ入力する。制御部23はEP_mapを作成する。

【0551】ステップS 267で、現在のパケットが最後に入力されるトランスポートパケットであるかどうかを判定する。最後のパケットでない場合、ステップS 262へ戻る。最後のパケットである場合、処理を終了する。

【0552】図144は、アナログAV信号をエンコードして記録する場合、図81に示すシンタクスのClipMarkの作成方法を説明するフローチャートを示す。

【0553】ステップS 281で、解析部14は端子11,12からの入力AV信号を解析して特徴点を検出する。特徴点はAVストリームの内容に起因する特徴的なシーンを指定する、例えば番組の頭だし点やシーンチェンジ点などである。

【0554】ステップS 282で、制御部23は特徴点の画像のPTSを取得する。

【0555】ステップS 283で、制御部23は特徴点の情報をClipMarkにストアする。具体的には、本実施例のClipMarkのシンタクスとセマンティクスで説明した情報をストアする。

【0556】図145は、デジタルインタフェースから入力されたトランスポートストリームを記録する場合、図81に示すシンタクスのClipMarkの作成方法を説明するフローチャートを示す。

【0557】ステップS 301で、デマルチプレクサ26および制御部23は記録するプログラムのエレメンタリストリームPIDの取得する。解析対象のエレメンタリストリームが複数ある場合は、すべてのエレメンタリストリームPIDを取得する。

【0558】ステップS 302で、デマルチプレクサ26は、端子13から入力されるトランスポートストリームのプログラムからエレメンタリストリームを分離し、それをAVデコーダ27へ入力する。AVデコーダ27は、入力ストリームをデコードしてAV信号を出力する。

【0559】ステップS 303で、解析部14は上記AV信号を解析して特徴点を検出する。

【0560】ステップS 304で、制御部23は特徴点

の画像のPTSとそれが属するSTCのSTC-sequence-idを取得する。

【0561】ステップS 305で、制御部23は特徴点の情報をClipMarkにストアする。具体的には、本実施例のClipMarkのシンタクスとセマンティクスで説明した情報をストアする。

【0562】次に、EP_mapを使用して特殊再生を行う方法を説明する。EP_mapはランダムアクセス再生のために役立つ。

【0563】デジタル放送のトランスポートストリームの中では、ビデオPIDが変化する場合がありますので、デコーダは記録されているトランスポートストリームの中のPIDマッピングを知る必要がある。そのために、EP_mapはEP_map_for_one_stream_PID()と呼ばれるサブテーブル毎にそれが参照するビデオPIDの値を持ち、また、ProgramInfoはPIDマッピングについての情報を持つ。

【0564】図146は、トランスポートストリームの中でビデオPIDの値が変化するDVR MPEG2 TSの例を示す。この場合、EP_mapはビデオPID毎にサブテーブルを持つ。

【0565】図147は、Iピクチャサーチ(トリックプレイ、チャプターサーチなど)をする場合のプレーヤモデルを示す。Iピクチャサーチは次のステップで行う。

【0566】1)最初に、ファイルシステム112はディスク111(図1の記録媒体100に対応する)からClip Information file(EP_map, STC_Info, ProgramInfo)のデータを読む、そのデータはホストコントローラ115へ送られる。

2)ユーザインタフェースは、再生するプログラム番号およびサーチ開始時間のPTSをセットする。そしてその値はホストコントローラ115へ送られる。

3)ホストコントローラ115は、サーチ開始時間に対応するところのRSPN_EP_startが指すソースパケットのビデオPIDをデマルチプレクサ113にセットする。

4)ホストコントローラ115は、前記RSPN_EP_startのソースパケット番号に対応するデータアドレスをファイルシステム112にセットする。

5)ファイルシステム112は、指定されたデータアドレスからDVR MPEG2トランスポートストリームを読み出す。

6)ユーザが次のサーチ時間をセットする場合は、上記ステップ2)へ戻る。

【0567】次に、図148は、オリジナルのAVストリームファイルと、そのストリームの部分的な再生範囲のストリームを消去する編集を行った後のAVストリームファイルの例を示す。

【0568】編集前に、Virtual Playlistは、オリジナルAVストリーム上のIN_timeとOUT_timeを指しているとする。この時、Virtual Playlistが使用していないスト

リーム部分を消去する編集（ミニマイズ編集）をした場合、それはオリジナルAVストリームを図148に示す編集後のストリームへ変える。オリジナルAVストリームの先頭からX点までのデータと、Y点から最後までまでのデータが消去される。以下の説明では、このX点とY点を決める方法の例を説明する。

【0569】図149は、AVストリームの内容を解析することをしないで、IN点の前の不要なデータを消去する方法を説明する図である。PlaylistはオリジナルAVストリーム上のIN点を指す。また、そのAVストリームのEP_mapを図示する。IN点が指すピクチャをデコードするためには、アドレスISA2から開始するIピクチャが必要である。

【0570】また、X点の後で、PAT、PMTおよびPCRパケットが必要である。RSPN_EP_start=ISA1のPTSはpts1であり、RSPN_EP_start=ISA2のPTSはpts2である。pts1とpts2のシステムタイムベースの時間差が100 msec以上ならば、アドレスISA1とISA2の間にはPAT、PMTおよびPCRパケットが存在する（少なくとも、SESF、DVB、ATSC、ISDBの場合はそうである）。

【0571】したがって、X点はアドレスISA1の前に決められる。そして、X点はアラインドユニットの境界でなければならない。

【0572】記録装置（図1の記録再生装置1）は、AVストリームの内容を解析することをしないで、X点をEP_mapを使用して次のステップで決めることができる。

【0573】1）システムタイムベース上でIN timeのPTSに最も近く、かつそれよりも過去の表示時刻のPTSの値を持つRSPN_EP_startを見つける。

2）ステップ1）で見つけたRSPN_EP_startのPTSの値よりも少なくとも100msec過去の表示時刻のPTSの値を持つRSPN_EP_startを見つける。

3）X点は、ステップ2）で見つけたRSPN_EP_startよりも前に決められる。そして、X点はアラインドユニットの境界でなければならない。

【0574】この方法は、X点を決めるためにAVストリームのデータを読み出し、その内容を解析することを必要としないので、簡単である。しかし、編集後のAVストリームは、そのPlaylistの再生には不要なデータを残してしまう場合がある。もし、X点を決めるためにAVストリームのデータを読み出し、その内容を解析するならば、そのPlaylistの再生には不要なデータをより効率良く消去できる。

【0575】図150は、AVストリームの内容を解析することをしないで、OUT点の後ろの不要なデータを消去する方法を説明する図である。PlaylistはオリジナルAVストリーム上のOUT点を指す。また、そのAVストリームのEP_mapを図示する。

【0576】RSPN_EP_start=ISA4から開始するビデオシーケンスは次に示すものであることを前提とする。

I2 B0 B1 P5 . . .

ここで、I、P、BはそれぞれIピクチャ、PピクチャそしてBピクチャを表す。数字は表示順序を表す。この処理において、記録装置がAVストリームの内容を解析しない場合、記録装置はOUT_timeのPTSが参照するところのピクチャの情報（ピクチャコーディングタイプ、テンポラル・レファレンスなど）がわからない。OUT_timeのPTSはピクチャB0またはB1を参照しているかもしれない（記録装置がAVストリームの内容を解析しない場合、このことはわからない）、この場合、ピクチャB0、B1をデコードするためにはI2が必要である。I2のPTSはOUT_timeのPTSよりも大きい（OUT_time < pts4、ここでpts4はI2のPTSである）。I2のPTSはOUT_timeのPTSよりも大きい、B0、B1のためにI2が必要である。

【0577】したがって、Y点は図に示すアドレスISA5の後ろに決められる。ISA5は、EP_mapの中でISA4の直後にあるRSPN_EP_startの値である。Y点はまたアラインドユニットの境界でなければならない。

【0578】記録装置は、AVストリームの内容を解析することをしないで、Y点をEP_mapを使用して次のステップで決めることができる。

【0579】1）システムタイムベース上でOUT_timeのPTSに最も近く、かつそれよりも未来の表示時刻のPTSの値を持つRSPN_EP_startを見つける。

2）ステップ1）で見つけたRSPN_EP_startの直後にあるRSPN_EP_startを見つける。

3）Y点は、ステップ2）で見つけたRSPN_EP_startよりも後ろに決められる。そして、Y点はアラインドユニットの境界でなければならない。

【0580】この方法は、Y点を決めるためにAVストリームのデータを読み出し、その内容を解析することを必要としないので、簡単である。しかし、編集後のAVストリームは、そのPlaylistの再生には不要なデータを残してしまう場合がある。もし、Y点を決めるためにAVストリームのデータを読み出し、その内容を解析するならば、そのPlaylistの再生には不要なデータをより効率良く消去できる。

【0581】このようなシンタクス、データ構造、規則に基づく事により、記録媒体に記録されているデータの内容、再生情報などを適切に管理することができ、もって、ユーザが再生時に適切に記録媒体に記録されているデータの内容を確認したり、所望のデータを簡便に再生できるようにすることができる。

【0582】本発明のデータベース構成によれば、PlaylistファイルやClip Informationファイルを別々に分離して記録するので、編集などによって、あるPlaylistやClipの内容を変更したとき、そのファイルに関係のない他のファイルを変更する必要がない。したがって、ファイルの内容の変更が容易に行え、またその変更および記録にかかる時間を小さくできる。

【0583】また、最初にInfo.dvrだけを読み出して、ディスクの記録内容をユーザインタフェースへ提示し、ユーザが再生指示したPlayListファイルとそれに関連するClip Informationファイルだけをディスクから読み出すようにすれば、ユーザの待ち時間を小さくすることができる。

【0584】もし、すべてのPlayListファイルやClip Informationファイルを1つのファイルにまとめて記録すると、そのファイルサイズは非常に大きくなる。そのために、そのファイルの内容を変更して、それを記録するためにかかる時間は、個々のファイルを別々に分離して記録する場合に比べて、非常に大きくなる。本発明は、この問題を解決する。

【0585】以上のように、AVストリームの付属情報として、前記AVストリームのClipInfoと、前記AVストリームのSTCの不連続点情報をストアするSTCInfoと、前記AVストリームのプログラム内容の不連続点情報をストアするProgramInfoと、前記AVストリーム上の時間情報とアドレス情報を関連づける情報をストアするためのCPIと、前記AVストリーム上の特徴的な画像のピクチャのタイプ（例えば番組の頭出し点）とそのピクチャのAVストリームの中のアドレスをストアするためのClipMarkを、Clip Information Fileとしてファイル化して記録媒体100に記録することにより、AVストリームの再生に必要なストリームの再生に必要なストリームの符号化情報を適切に管理することが可能である。

【0586】このClip Information file情報により、ユーザが、記録媒体100に記録されているAVストリームの中から興味のあるシーン、例えば番組の頭出し点など、をサーチすることができ、ユーザのランダムアクセスや特殊再生の指示に対して、記録媒体100からのAVストリームの読み出し位置の決定が容易になり、またストリームの復号開始を速やかに行うことができる。

【0587】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。この場合、例えば、記録再生装置1は、図151に示されるようなパーソナルコンピュータにより構成される。

【0588】図151において、CPU (Central Processing Unit) 201は、ROM (Read Only Memory) 202に記憶されているプログラム、または記憶部208からRAM (Random Access Memory) 203にロードされたプログラムに従って各種の処理を実行する。RAM 203にはまた、CPU 201が各種の処理を実行する上において必要なデータなども適宜記憶される。

【0589】CPU 201、ROM 202、およびRAM 203は、バス204を介して相互に接続されている。このバス204にはまた、入出力インタフェース205も接続されている。

【0590】入出力インタフェース205には、キーボ

ード、マウスなどよりなる入力部206、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部207、ハードディスクなどより構成される記憶部208、モデム、ターミナルアダプタなどより構成される通信部209が接続されている。通信部209は、ネットワークを介しての通信処理を行う。

【0591】入出力インタフェース205にはまた、必要に応じてドライブ210が接続され、磁気ディスク221、光ディスク222、光磁気ディスク223、或いは半導体メモリ224などが適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部208にインストールされる。

【0592】上述した一連の処理は、ハードウェアにより実行させることもできるが、ソフトウェアにより実行させることもできる。一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが専用のハードウェアに組み込まれているコンピュータ、または、各種のプログラムをインストールすることで、各種の機能を実行することが可能な、例えば汎用のパーソナルコンピュータなどに、記録媒体からインストールされる。

【0593】この記録媒体は、図151に示すように、コンピュータとは別に、ユーザにプログラムを提供するために配布される、プログラムが記録されている磁気ディスク221（フロッピーディスクを含む）、光ディスク222（CD-ROM (Compact Disk-Read Only Memory)、DVD (Digital Versatile Disk)を含む）、光磁気ディスク223（MD (Mini-Disk)を含む）、若しくは半導体メモリ224などよりなるパッケージメディアにより構成されるだけでなく、コンピュータに予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM 202や記憶部208が含まれるハードディスクなどで構成される。

【0594】なお、本明細書において、媒体により提供されるプログラムを記述するステップは、記載された順序に従って、時系列的に行われる処理は勿論、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0595】また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【0596】

【発明の効果】以上の如く、本発明の第1の情報処理装置および方法、記録媒体、並びにプログラムによれば、Clip情報として、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が記録される。

【0597】本発明の第2の情報処理装置および方法、記録媒体、並びにプログラムによれば、Clip情報とし

て、AVストリーム中の符号化情報が連続な区間の開始アドレス情報、AVストリーム中の時刻情報とアドレス情報を関連づける情報、およびAVストリーム中の特徴的な画像の時刻情報が再生される。

【0598】従って、いずれの場合においても、AVストリームの読み出し位置の決定や復号処理を速やかに行うことができ、特に、所定のマークを迅速に検索することができる。

【図面の簡単な説明】

【図1】本発明を適用した記録再生装置の一実施の形態の構成を示す図である。

【図2】記録再生装置1により記録媒体に記録されるデータのフォーマットについて説明する図である。

【図3】Real PlaylistとVirtual Playlistについて説明する図である。

【図4】Real Playlistの作成について説明する図である。

【図5】Real Playlistの削除について説明する図である。

【図6】アセンブル編集について説明する図である。

【図7】Virtual Playlistにサブパスを設ける場合について説明する図である。

【図8】Playlistの再生順序の変更について説明する図である。

【図9】Playlist上のマークとClip上のマークについて説明する図である。

【図10】メニューサムネイルについて説明する図である。

【図11】Playlistに付加されるマークについて説明する図である。

【図12】クリップに付加されるマークについて説明する図である。

【図13】Playlist、Clip、サムネイルファイルの関係について説明する図である。

【図14】ディレクトリ構造について説明する図である。

【図15】info.dvrのシンタクスを示す図である。

【図16】DVR volumeのシンタクスを示す図である。

【図17】Resumevolumeのシンタクスを示す図である。

【図18】UIAppInfovolumeのシンタクスを示す図である。

【図19】Character set valueのテーブルを示す図である。

【図20】TableOfPlaylistのシンタクスを示す図である。

【図21】TableOfPlaylistの他のシンタクスを示す図である。

【図22】MakersPrivateDataのシンタクスを示す図である。

【図23】xxxxx.rplsとyyyyy.vplsのシンタクスを示す

図である。

【図24】Playlistについて説明する図である。

【図25】Playlistのシンタクスを示す図である。

【図26】Playlist_typeのテーブルを示す図である。

【図27】UIAppinfoPlaylistのシンタクスを示す図である。

【図28】図27に示したUIAppinfoPlaylistのシンタクス内のフラグについて説明する図である。

【図29】PlayItemについて説明する図である。

【図30】PlayItemについて説明する図である。

【図31】PlayItemについて説明する図である。

【図32】PlayItemのシンタクスを示す図である。

【図33】IN_timeについて説明する図である。

【図34】OUT_timeについて説明する図である。

【図35】Connection_Conditionのテーブルを示す図である。

【図36】Connection_Conditionについて説明する図である。

【図37】BridgeSequenceInfoを説明する図である。

【図38】BridgeSequenceInfoのシンタクスを示す図である。

【図39】SubPlayItemについて説明する図である。

【図40】SubPlayItemのシンタクスを示す図である。

【図41】SubPath_typeのテーブルを示す図である。

【図42】PlaylistMarkのシンタクスを示す図である。

【図43】Mark_typeのテーブルを示す図である。

【図44】Mark_time_stampを説明する図である。

【図45】zzzzz.clipのシンタクスを示す図である。

【図46】ClipInfoのシンタクスを示す図である。

【図47】Clip_stream_typeのテーブルを示す図である。

【図48】offset_SPNについて説明する図である。

【図49】offset_SPNについて説明する図である。

【図50】STC区間について説明する図である。

【図51】STC_Infoについて説明する図である。

【図52】STC_Infoのシンタクスを示す図である。

【図53】ProgramInfoを説明する図である。

【図54】ProgramInfoのシンタクスを示す図である。

【図55】VideoCondInfoのシンタクスを示す図である。

【図56】Video_formatのテーブルを示す図である。

【図57】frame_rateのテーブルを示す図である。

【図58】display_aspect_ratioのテーブルを示す図である。

【図59】AudioCondInfoのシンタクスを示す図である。

【図60】audio_codingのテーブルを示す図である。

【図61】audio_component_typeのテーブルを示す図である。

【図62】sampling_frequencyのテーブルを示す図であ

る。

【図63】CPIについて説明する図である。

【図64】CPIについて説明する図である。

【図65】CPIのシンタクスを示す図である。

【図66】CPI_typeのテーブルを示す図である。

【図67】ビデオEP_mapについて説明する図である。

【図68】EP_mapについて説明する図である。

【図69】EP_mapについて説明する図である。

【図70】EP_mapのシンタクスを示す図である。

【図71】EP_type valuesのテーブルを示す図である。

【図72】EP_map_for_one_stream_PIDのシンタクスを示す図である。

【図73】TU_mapについて説明する図である。

【図74】TU_mapのシンタクスを示す図である。

【図75】ClipMarkのシンタクスを示す図である。

【図76】mark_typeのテーブルを示す図である。

【図77】mark_type_stampのテーブルを示す図である。

【図78】ClipMarkのシンタクスの他の例を示す図である。

【図79】Mark_typeのテーブルの他の例を示す図である。

【図80】mark_entry()とrepresentative_picture_entry()の例を示す図である。

【図81】mark_entry()とrepresentative_picture_entry()のシンタクスを示す図である。

【図82】mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

【図83】RSPN_ref_EP_startとoffset_num_picturesの関係を説明する図である。

【図84】mark_entry()とrepresentative_picture_entry()のシンタクスの他の例を示す図である。

【図85】ClipMarkとEP_mapの関係を説明する図である。

【図86】menu.thmbとmark.thmbのシンタクスを示す図である。

【図87】Thumbnailのシンタクスを示す図である。

【図88】thumbnail_picture_formatのテーブルを示す図である。

【図89】tn_blockについて説明する図である。

【図90】DVR MPEG 2のトランスポートストリームの構造について説明する図である。

【図91】DVR MPEG 2のトランスポートストリームのレコーダモデルを示す図である。

【図92】DVR MPEG 2のトランスポートストリームのプレーヤモデルを示す図である。

【図93】source packetのシンタクスを示す図である。

【図94】TP_extra_headerのシンタクスを示す図である。

【図95】copy permission indicatorのテーブルを示す図である。

【図96】シームレス接続について説明する図である。

【図97】シームレス接続について説明する図である。

【図98】シームレス接続について説明する図である。

【図99】シームレス接続について説明する図である。

【図100】シームレス接続について説明する図である。

【図101】オーディオのオーバーラップについて説明する図である。

【図102】BridgeSequenceを用いたシームレス接続について説明する図である。

【図103】BridgeSequenceを用いないシームレス接続について説明する図である。

【図104】DVR STDモデルを示す図である。

【図105】復号、表示のタイミングチャートを示す図である。

【図106】図81のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図107】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図81のシンタクスの場合における再生の動作を説明する図である。

【図108】EP_mapの例を示す図である。

【図109】ClipMarkの例を示す図である。

【図110】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

【図111】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図81のシンタクスの場合におけるCMスキップ再生処理を説明するフローチャートである。

【図112】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図82のシンタクスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図113】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図82のシンタクスの場合における再生を説明する図である。

【図114】EP_mapの例を示す図である。

【図115】ClipMarkの例を示す図である。

【図116】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図117】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図82のシンタクスの場合におけるCMスキップ再生を説明するフローチャートである。

【図118】図75または図78のClipMarkのmark_ent

ry()/representative_picture_entry()が図84のシンタックスの場合におけるマーク点で示されるシーンの頭出し再生を説明するフローチャートである。

【図119】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図84のシンタックスの場合における再生を説明する図である。

【図120】EP_mapの例を示す図である。

【図121】ClipMarkの例を示す図である。

【図122】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図84のシンタックスの場合におけるCMスキップ再生を説明するフローチャートである。

【図123】図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図84のシンタックスの場合におけるCMスキップ再生を説明するフローチャートである。

【図124】アプリケーションフォーマットを示す図である。

【図125】PlayList上のマークとClip上のマークを説明する図である。

【図126】ClipMarkのシンタックスの他の例を示す図である。

【図127】ClipMarkのシンタックスのさらに他の例を示す図である。

【図128】ClipInfo()のシンタックスの別の例を示す図である。

【図129】ProgramInfo()のシンタックスの別の例を示す図である。

【図130】StreamCodingInfo()のシンタックスを示す図である。

【図131】stream_coding_typeを説明する図である。

【図132】EP-fineとEP-coarseの関係を説明する図である。

【図133】PTS_EP_fineとPTS_EP_coarseのフォーマットを説明する図である。

【図134】RSPN_EP_fineとRSPN_EP_coarseのフォーマットを説明する図である。

【図135】EP-coarseのエントリーとEP-fineのエントリーを説明する図である。

【図136】EP_mapのシンタックスの別の例を示す図である。

【図137】EP_stream_type valuesを説明する図である。

【図138】図136のEP_mapのEP_map_for_one_strea

m_PIDのシンタックスを示す図である。

【図139】EP_video_typeの値の意味を説明する図である。

【図140】Clip AVストリームファイルおよびClip Informationファイルの作成処理を説明するフローチャートである。

【図141】STC_Infoの作成の動作例を説明するフローチャートである。

【図142】ProgramInfoの作成の動作例を説明するフローチャートである。

【図143】EP_mapの作成の動作例を説明するフローチャートである。

【図144】アナログAV信号をエンコードして記録する場合における、図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図81に示すシンタックスであるときのClipMarkの作成方法を説明するフローチャートである。

【図145】デジタルインタフェースから入力されたトランスポートストリームを記録する場合における、図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図81に示すシンタックスであるときのClipMarkの作成方法を説明するフローチャートである。

【図146】EP_mapを使う特殊再生を説明する図である。

【図147】EP_mapを使用したIピクチャサーチのためのプレーヤモデルを説明する図である。

【図148】ミニマイズのオペレーションの例を示す図である。

【図149】ミニマイズのときにIN_timeの前の不要なストリームデータを消去する例を示す図である。

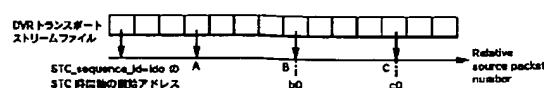
【図150】ミニマイズのときにOUT_timeの後ろの不要なストリームデータを消去する例を説明する図である。

【図151】媒体を説明する図である。

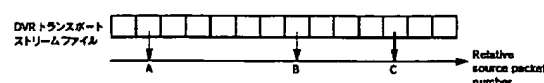
【符号の説明】

1 記録再生装置, 11乃至13 端子, 14 解析部, 15 AVエンコーダ, 16 マルチプレкса, 17 スイッチ, 18 多重化ストリーム解析部, 19 ソースパケットタイザ, 20 ECC符号化部, 21 変調部, 22 書き込み部, 23 制御部, 24 ユーザインタフェース, 26 デマルチプレкса, 27 AVデコーダ, 28 読み出し部, 29復調部, 30 ECC復号部, 31 ソースパケットタイザ, 32, 33 端子

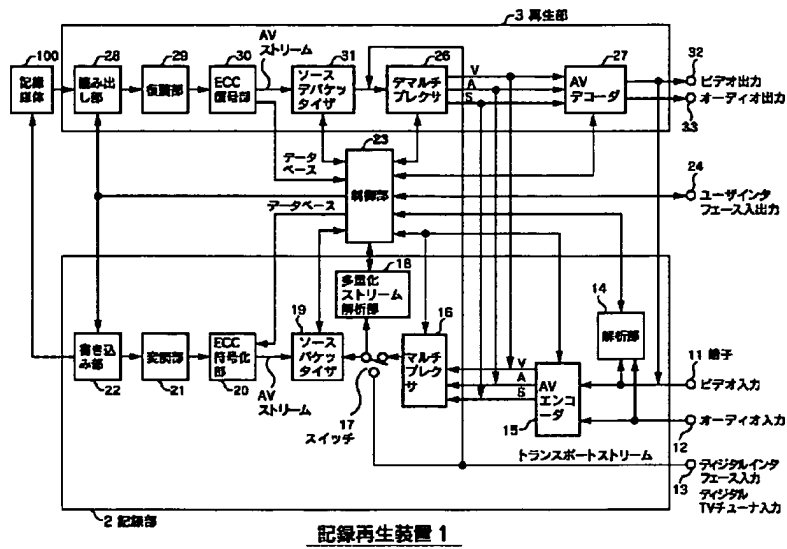
【図107】



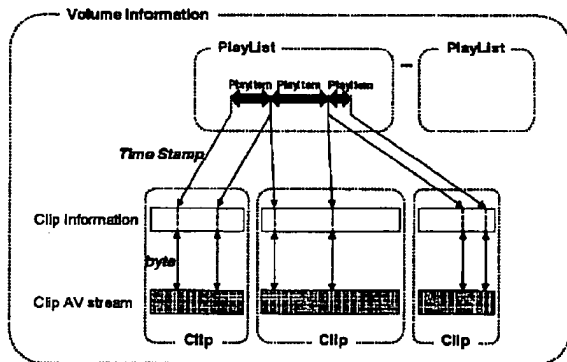
【図113】



【図1】

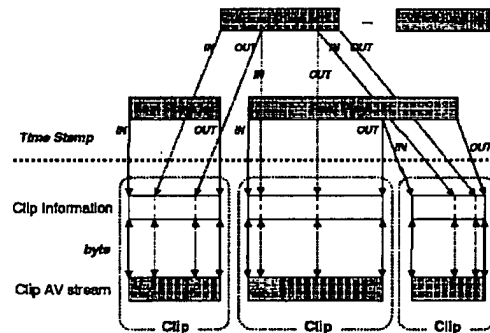


【図2】

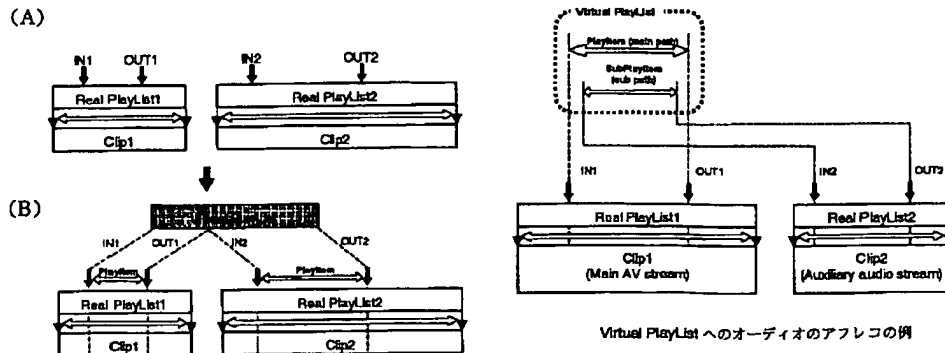


【図6】

【図3】

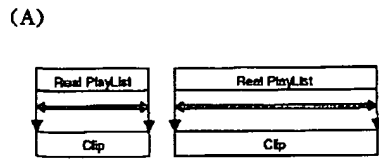


【図7】

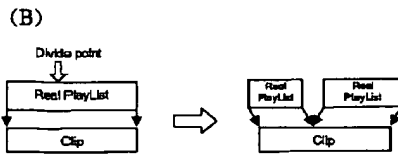


Virtual PlayList へのオーディオのアフレコの例

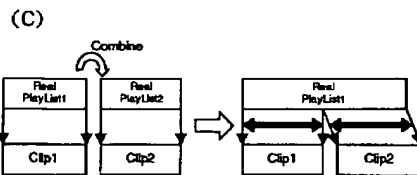
【図4】



Real Playlistのクリエイトの例

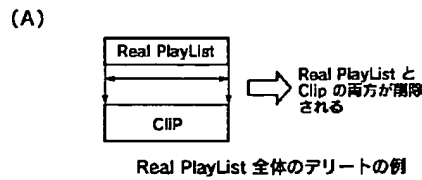


Real Playlistのディバイドの例

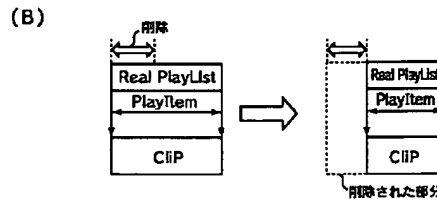


Real Playlistのコンバインの例

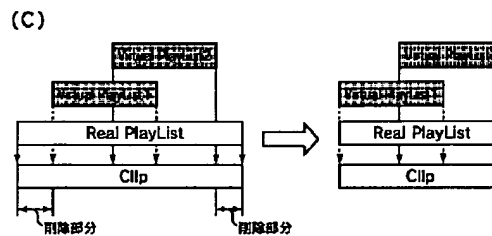
【図5】



Real Playlist 全体のデリートの例



Real Playlist の部分的なデリートの例



Real Playlist のミニマイズの例

【図108】

EP_map

RSPN_EP_start	PTS_EP_start
...	...
A	PTS (A)
B	PTS (B)
C	PTS (C)
...	...

【図114】

EP_map

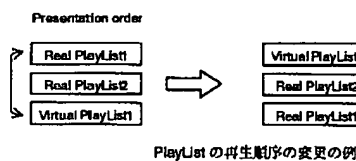
RSPN_EP_start	PTS_EP_start
...	...
A	PTS (A)
B	PTS (B)
C	PTS (C)
...	...

【図120】

EP_map

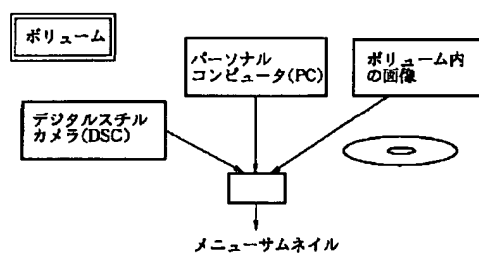
RSPN_EP_start	PTS_EP_start
...	...
A	PTS (A)
B	PTS (B)
C	PTS (C)
...	...

【図8】



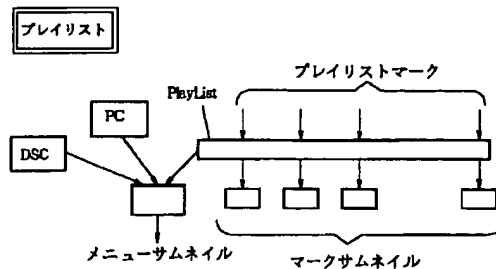
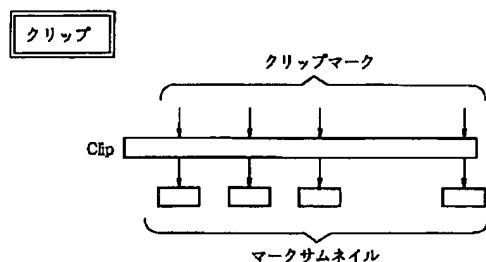
Playlistの再生順序の変更の例

【図10】

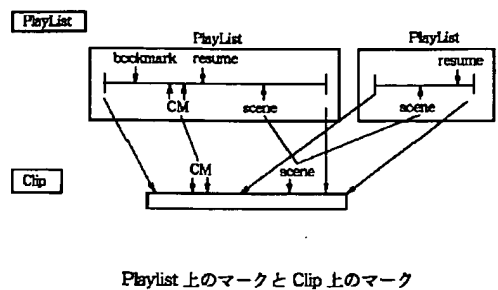


【図11】

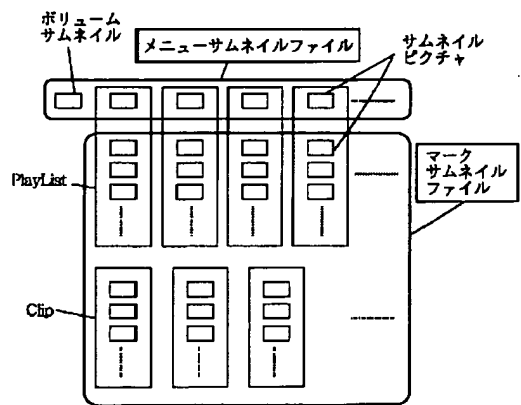
【図12】



【 図 9 】



【 図 1 3 】

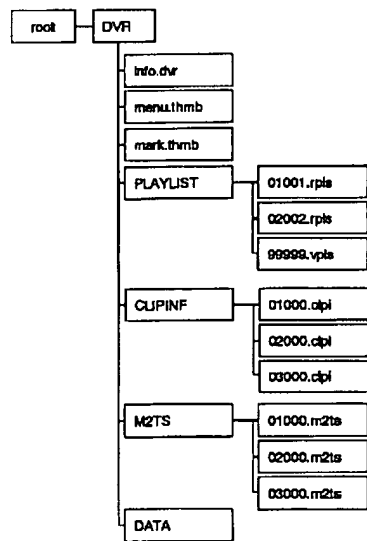


【 図 1 6 】

Syntax	No. bits	of Mnemonics
DVRVolume() {		
version_number	8*4	bslbf
length	32	uimsbf
ResumeVolume()		
UAppInfoVolume()		
}		

DVR Volume のシンタクス

【 図 1 4 】



【 図 1 5 】

Syntax	No. bits	of Mnemonics
Info.dvr {		
TableOfPlayLists_Start_address	32	uimsbf
MakersPrivateData_Start_address	32	uimsbf
reserved	192	bslbf
DVRVolume()		
for(i=0;i<N1;i++){		
padding_word	16	bslbf
}		
TableOfPlayLists()		
for(i=0;i<N2;i++){		
padding_word	16	bslbf
}		
MakersPrivateData()		
}		

info.dvr のシンタクス

【 図 1 9 】

Value	Character coding
0x00	Reserved
0x01	ISO/IEC 648 (ASCII)
0x02	ISO/IEC 10646-1 (Unicode)
0x03-0xff	Reserved

Character set value

【図17】

Syntax	No. bits	of	Mnemonics
ResumeVolume() {			
reserved	15		bslbf
valid flag	1		bslbf
resume PlayList name	8*10		bslbf
}			

ResumeVolume のシンタクス

【図18】

Syntax	No. bits	of	Mnemonics
UIAppInfoVolume () {			
character set	8		bslbf
name length	8		uimsbf
Volume name	8*256		bslbf
reserved	15		bslbf
Volume protect flag	1		bslbf
PIN	8*4		bslbf
ref thumbnail index	16		uimsbf
reserved for future use	256		bslbf
}			

UIAppInfoVolume のシンタクス

【図20】

Syntax	No. bits	of	Mnemonics
TableOfPlayLists() {			
version number	8*4		bslbf
length	32		uimsbf
number of PlayLists	16		uimsbf
for (i=0; i<number of PlayLists; i++) {			
PlayList file name	8*10		bslbf
}			
}			

TableOfPlayLists のシンタクス

【図21】

■ TableOfPlayLists - シンタクス (4.2.3.2 の別案)

Syntax	No. bits	of	Mnemonics
TableOfPlayLists() {			
version number	8*4		bslbf
length	32		uimsbf
number of PlayLists	16		uimsbf
for (i=0; i<number of PlayLists; i++) {			
PlayList file name	8*10		bslbf
UIAppInfoPlayList()			
}			
}			

TableOfPlayLists の別シンタクス

【図23】

Syntax	No. bits	of	Mnemonics
xxxx.rpls / yyyy.vpls {			
PlayListMark_Start address	32		uimsbf
MarkersPrivateData_Start address	32		uimsbf
reserved	192		bslbf
PlayList()			
for (i=0; i<N1; i++) {			
padding_word	16		bslbf
}			
PlayListMark()			
for (i=0; i<N2; i++) {			
padding_word	16		bslbf
}			
MarkersPrivateData()			
}			

xxxx.rpls と yyyy.vpls のシンタクス

【図26】

PlayList_type	Meaning
0	AV記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のビデオストリームを含まなければならない。
1	オーディオ記録のためのPlayList このPlayListに参照されるすべてのClipは、一つ以上のオーディオストリームを含まなければならない、そしてビデオストリームを含んではならない。
2-255	reserved

PlayList_type

【図28】

(A)

write_protect_flag	Meaning
0b	そのPlayListを自由に消去しても良い。
1b	write_protect_flagを除いてそのPlayListの内容は、消去および変更されるべきではない。

write_protect_flag

(B)

is_played_flag	Meaning
0b	そのPlayListは、記録されてから一度も再生されたことがない。
1b	PlayListは、記録されてから一度は再生された。

is_played_flag

(C)

archive	Meaning
00b	何も情報が定義されていない。
01b	オリジナル
10b	コピー
11b	reserved

archive

【図22】

Syntax	No. bits	of Mnemonics
MakersPrivateData() {		
version number	8*4	bslbf
length	32	uimbsf
if (length != 0) {		
mpd blocks start address	32	uimbsf
number of maker entries	16	uimbsf
mpd block size	16	uimbsf
number of mpd blocks	16	uimbsf
reserved	16	bslbf
for (i=0; i<number of maker entries; i++) {		
maker ID	16	uimbsf
maker model code	16	uimbsf
start mpd block number	16	uimbsf
reserved	16	bslbf
mpd length	32	uimbsf
}		
stuffing bytes	8*2*L1	bslbf
for (i=0; i<number of mpd blocks; i++) {		
mpd_block	mpd_block size*1024*8	
}		
}		

MakersPrivateData のシンタクス

【図25】

Syntax	No. bits	of Mnemonics
Playlist() {		
version number	8*4	bslbf
length	32	uimbsf
Playlist type	8	uimbsf
CPI type	1	bslbf
reserved	7	bslbf
UIAppInfoPlaylist() {		
number of PlayItems // main path	16	uimbsf
if (<Virtual Playlist>) {		
number of SubPlayItems // sub path	16	uimbsf
} else {		
reserved	16	bslbf
}		
for (PlayItem_id=0; PlayItem_id<number of PlayItems; PlayItem_id++) {		
PlayItem() // main path		
}		
if (<Virtual Playlist>) {		
if (CPI_type==0 && Playlist_type==0) {		
for (i=0; i<number of SubPlayItems; i++) {		
SubPlayItem() // sub path		
}		
}		
}		

Playlist のシンタクス

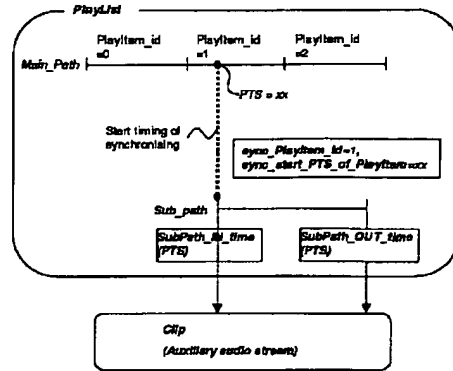
【図33】

CPI_type in the Playlist()	Semantics of IN_time
EP_map type	IN_time は、PlayItem の中で最初のプレゼンテーションユニットに対応する33ビット長のPTSの上位32ビットを示さなければならない。かつ、IN_time は、time_unitの精度に丸めて表さなければならない。IN_time は、次に示す等式により計算される。
TU_map type	

$$IN_time = TU_start_time \% 2^{32}$$

IN_time

【図39】



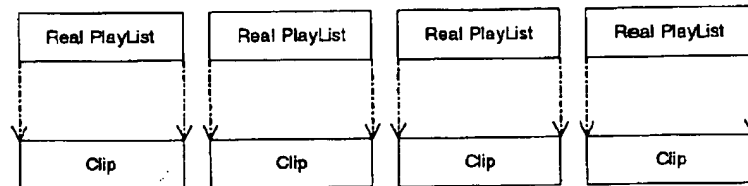
【図41】

SubPath_type	Meaning
0x00	Auxiliary audio stream path
0x01 - 0xff	reserved

SubPath_type

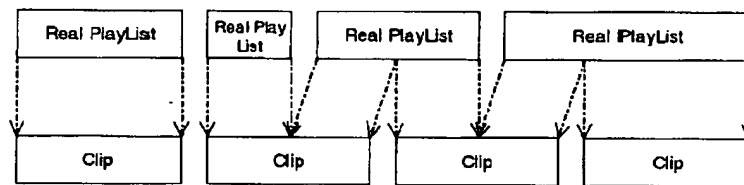
【図24】

(A)



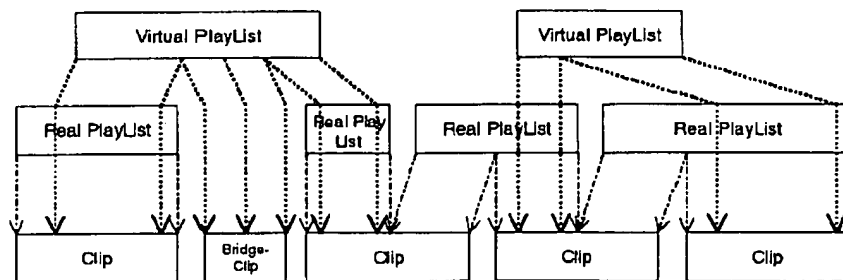
初めて AV ストリームが Clip として記録された時の Real PlayList の例

(B)



編集後の Real PlayList の例

(C)



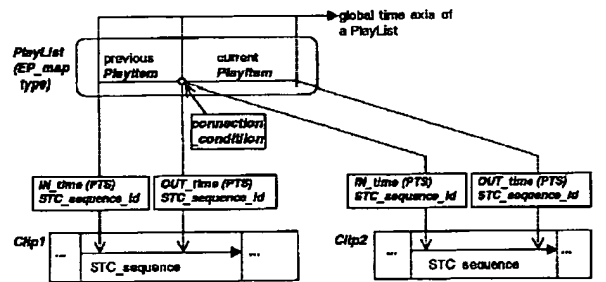
Virtual PlayList の例

【図27】

Syntax	No. of bits	of Mnemonics
UIAppInfoPlaylist2() {		
character set	8	bsbf
name length	8	uimbsf
Playlist name	8*256	bsbf
reserved	8	bsbf
record time and date	4*14	bsbf
reserved	8	bsbf
duration	4*8	bsbf
valid period	4*8	bsbf
maker id	16	uimbsf
maker code	16	uimbsf
reserved	11	bsbf
playback control flag	1	bsbf
write protect flag	1	bsbf
is played flag	1	bsbf
archive	2	bsbf
ref thumbnail index	16	uimbsf
reserved for future use	256	bsbf
}		

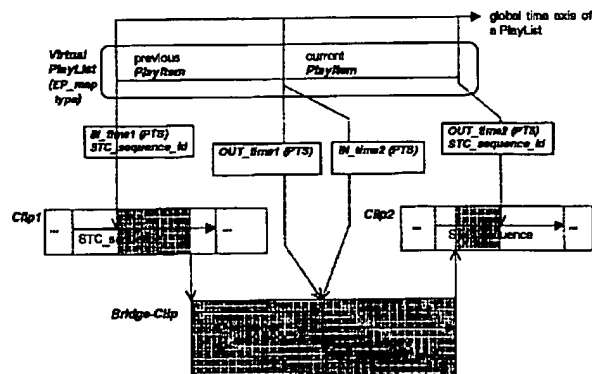
UIAppInfoPlaylist のシNTAX

【図29】



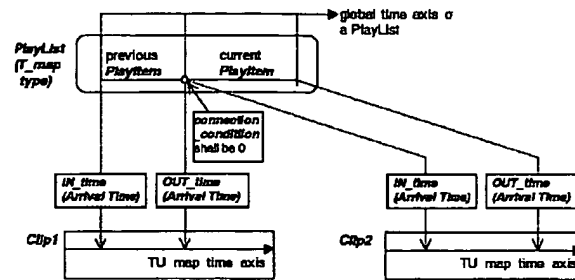
Playlist が EP_map type であり、かつ PlayItem が BridgeSequence を持たない時の例

【図30】



Playlist が EP_map type であり、かつ PlayItem が BridgeSequence を持つ時の例

【図31】



PlayList が TU_map type である時の例

【図32】

Syntax	No. bits	Minemonics
PlayItem() {		
Clip information file name	8*10	bslbf
reserved	24	bslbf
STC sequence Id	8	uimbsf
IN time	32	uimbsf
OUT time	32	uimbsf
reserved	14	bslbf
connection condition	2	bslbf
if (cVirtual PlayList) {		
if (connection condition == '10') {		
BridgeSequenceInfo()		
}		
}		
}		

PlayItem のシンタクス

【図34】

CPI_type in the PlayList()	Semantics of OUT_time
EP_map type	<p>OUT_time は、次に示す等式によって計算される Presentation_end_TS の値の上位 32 ビットを示さなければならない。</p> $Presentation_end_TS = PTS_out + AU_duration$ <p>ここで、</p> <p>PTS_out は、PlayItem の中で最後のプレゼンテーションユニットに対応する 32 ビット長の PTS である。</p> <p>AU_duration は、最後のプレゼンテーションユニットの 80kHz 単位の表示期間である。</p>
TU_map type	<p>OUT_time は、TU_map_time_axis 上の時刻でなければならない。かつ、OUT_time は、time_unit の精度に丸めて表さなければならない。</p> <p>OUT_time は、次に示す等式により計算される。</p> $OUT_time = TU_start_time \% 2^{22}$

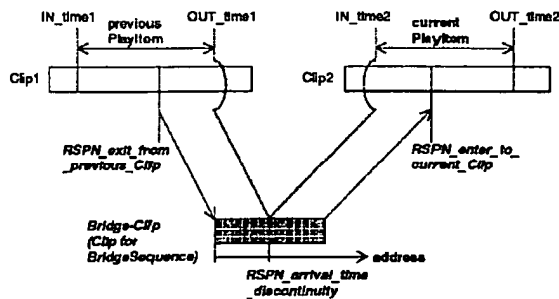
OUT_time

【図35】

connection condition	meaning
00	<ul style="list-style-type: none"> 先行する PlayItem と現在の PlayItem との接続は、シームレス再生の保証がなされていない。 PlayList の CPI_type が TU_map type である場合 connection_condition は、この値をセットされねばならない。
01	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、システムタイムベース（STC ベース）の不連続点があるために分割されていることを表す。
10	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 この状態は、Virtual PlayList に対してだけ許される。 先行する PlayItem と現在の PlayItem との接続は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用して接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていなければならない。
11	<ul style="list-style-type: none"> この状態は、PlayList の CPI_type が EP_map type である場合にだけ許される。 先行する PlayItem と現在の PlayItem は、シームレス再生の保証がなされている。 先行する PlayItem と現在の PlayItem は、BridgeSequence を使用しないで接続されており、DVR MPEG-2 トランスポートストリームは、後述する DVR-STD に従っていなければならない。

connection_condition

【図37】

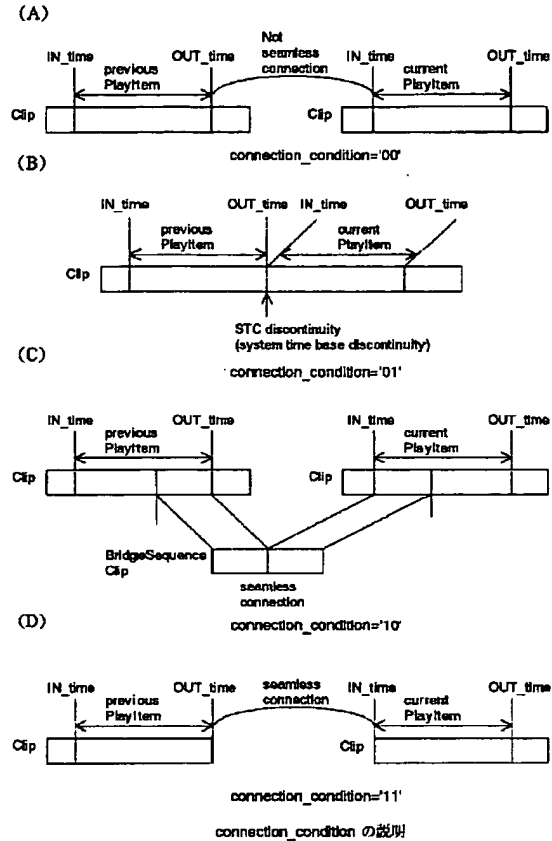


【图47】

Clip stream type	meaning
0	Clip AV ストリーム
1	Bridge-Clip AV ストリーム
2 - 255	Reserved

Clip_stream_type

【図36】



【図44】

CPI type in the Playlist	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表わなければならない。mark_time_stamp は、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

mark time stamp

【 38 】

Syntax	No. of bits	Mnemonics
BridgeSequenceInfo {		
Bridge Clip Information file name	8*10	bsbf
RSPM exit from previous Clip	32	uimsbf
RSPM enter to current Clip	32	uimsbf
}		

BridgeSequenceInfo のシンタクス

【 40 】

Syntax	No. of bits	Mnemonics
SubPlayItem() {		
Clip Information file name	8*10	bsbf
SubPath type	8	bsbf
sync PlayItem id	8	uimsbf
sync start PTS of PlayItem	32	uimsbf
SubPath IN time	32	uimsbf
SubPath OUT time	32	uimsbf
}		

SubPlayItem のシンタクス

【 56 】

video_format	Meaning
0	480i
1	576i
2	480p (including 640x480p format)
3	1080i
4	720p
5	1080p
6-254	reserved
255	No information

video_format

【 42 】

Syntax	No. of bits	Mnemonics
PlayListMark() {		
version number	8*4	bsbf
length	32	uimsbf
number of PlayList marks	16	uimsbf
for(i=0; i < number of PlayList marks; i++) {		
reserved	8	bsbf
mark type	8	bsbf
mark time stamp	32	uimsbf
PlayItem id	8	uimsbf
reserved	24	uimsbf
character set	8	bsbf
name length	8	uimsbf
mark name	8*256	bsbf
ref thumbnail index	16	uimsbf
}		
}		

PlayListMark のシンタクス

【図43】

Mark type	Meaning	Comments
0x00	resume-mark	再生リジュームポイント。PlayListMark()において 定義される再生リジュームポイントの数は、0または 1でなければならない。
0x01	book-mark	PlayListの再生エントリーポイント。このマークは、 ユーザがセットすることができ、例えば、お気に入りの シーンの開始点を指定するマークに使う。
0x02	skip-mark	スキップマークポイント。このポイントからプログ ラムの最後まで、プレーヤはプログラムをスキップ する。PlayListMark() において定義されるスキップ マークポイントの数は、0または1でなければなら ない。
0x03 - 0xBF	reserved	
0x90 - 0xFF	reserved	Reserved for ClipMark()

mark_type

【図45】

Syntax	No. of bits	of	Mnemonics
zzzz.clp {			
STC_Info_Start_address	32	uint32	
ProgramInfo_Start_address	32	uint32	
CPI_Start_address	32	uint32	
ClipMark_Start_address	32	uint32	
MakersPrivateData_Start_address	32	uint32	
reserved	96	bits	
ClipInfo()			
for (i=0; i<N1; i++){			
padding_word	16	bits	
}			
STC_Info()			
for (i=0; i<N2; i++){			
padding_word	16	bits	
}			
ProgramInfo()			
for (i=0; i<N3; i++){			
padding_word	16	bits	
}			
CPI()			
for (i=0; i<N4; i++){			
padding_word	16	bits	
}			
ClipMark()			
for (i=0; i<N5; i++){			
padding_word	16	bits	
}			
MakersPrivateData()			
}			

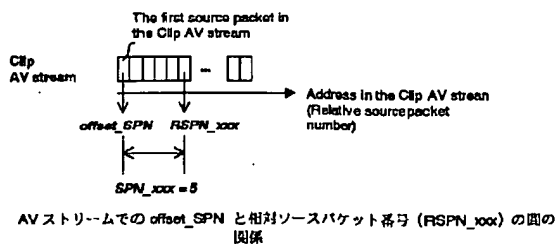
zzzz.clpのシンタクス

【図46】

Syntax	No. of bits	of	Mnemonics
ClipInfo {			
version number	8	bits	
length	32	uint32	
clip stream type	8	bits	
offset_SPN	32	uint32	
TS recording rate	24	uint32	
reserved	8	bits	
record time and date	4*14	bits	
reserved	8	bits	
duration	4*8	bits	
reserved	7	bits	
time controlled flag	1	bits	
TS average rate	24	uint32	
// (Clip stream type==1) // Bridge-Clip AV stream			
RSPN arrival time discontinuity	32	uint32	
else reserved	32	bits	
reserved for system use	144	bits	
reserved	11	bits	
is format identifier valid	1	bits	
is original network ID valid	1	bits	
is transport stream ID valid	1	bits	
is service ID valid	1	bits	
is country code valid	1	bits	
format identifier	32	bits	
original network ID	16	uint16	
transport stream ID	16	uint16	
service ID	16	uint16	
country code	24	bits	
stream format name	16*8	bits	
reserved for future use	256	bits	
}			

ClipInfoのシンタクス

【図49】

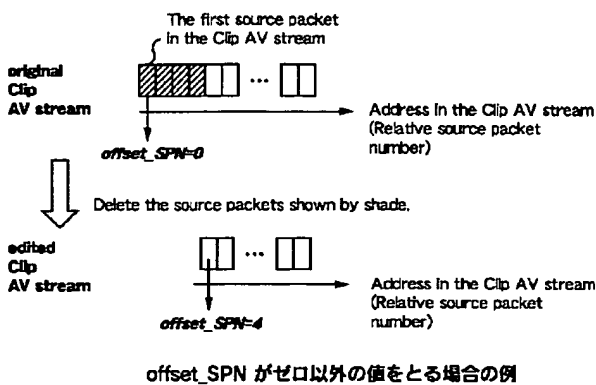


【図57】

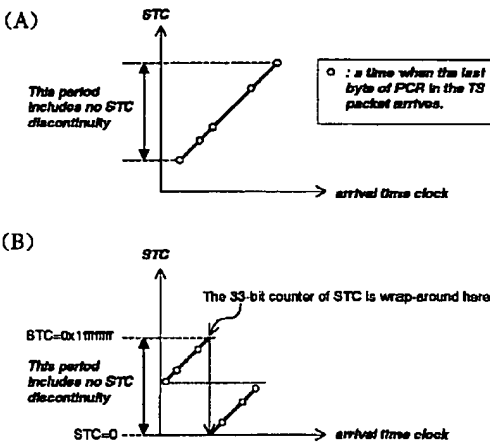
frame_rate	Meaning
0	forbidden
1	24 000/1001 (23.976...)
2	24
3	25
4	30 000/1001 (29.97...)
5	30
6	50
7	60 000/1001 (59.94...)
8	60
9 - 254	reserved
255	No information

frame_rate

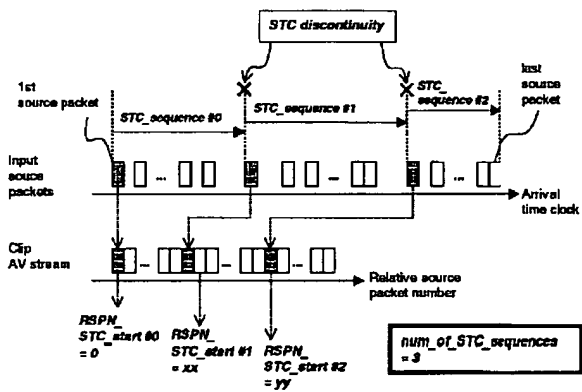
【 図 48 】



【 図 50 】



【 図 51 】



-STC_Info

【 図 66 】

CPI_type	Meaning
0	EP map type
1	TU map type

CPI_type の意味

【 図 52 】

Syntax	No. of bits	of	Minamonic
STC_Info {			
version number	8*4		bslbf
length	32		ulmsbf
if (length != 0) {			
reserved	8		bslbf
num_of_STC_sequences	8		ulmsbf
for(STC_sequence_id=0; STC_sequence_id < num_of_STC_sequences; STC_sequence_id++) {			
reserved	32		bslbf
RSPN STC_start	32		ulmsbf
}			
}			

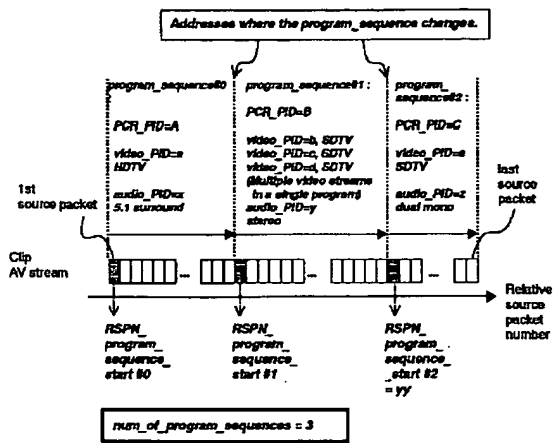
STC_Info のシンタクス

【 図 61 】

audio_component_type	Meaning
0	single mono channel
1	dual mono channel
2	stereo (2-channel)
3	multi-lingual, multi-channel
4	surround sound
5	audio description for the visually impaired
6	audio for the hard of hearing
7-254	reserved
255	No information

audio_component_type

【 53 】



ProgramInfo の例

【 58 】

display_aspect_ratio	Meaning
0	forbidden
1	reserved
2	4:3 display aspect ratio
3	16:9 display aspect ratio
4-254	reserved
255	No information

display_aspect_ratio

【 54 】

Syntax	No. of bits	of	Mnemonics
ProgramInfo {			
version number	8*4		bslbf
length	32		uimsbf
if (length != 0) {			
reserved	8		bslbf
number of program sequences	8		uimsbf
for (i=0; i<number of program sequences; i++) {			
RSPN program sequence start	32		uimsbf
reserved	48		bslbf
PCR PID	16		bslbf
number of videos	8		uimsbf
number of audios	8		uimsbf
for (k=0; k<number of videos; k++) {			
video stream PID	16		bslbf
VideoCodingInfo {			
}			
for (k=0; k<number of audios; k++) {			
audio stream PID	16		bslbf
AudioCodingInfo {			
}			
}			
}			

ProgramInfo のシンタクス

【 62 】

sampling_frequency	Meaning
0	48 kHz
1	44.1 kHz
2	32 kHz
3-254	reserved
255	No information

sampling_frequency

【 55 】

Syntax	No. of bits	of	Mnemonics
VideoCodingInfo {			
video format	8		uimsbf
frame rate	8		uimsbf
display_aspect_ratio	8		uimsbf
reserved	8		bslbf
}			

VideoCodingInfo のシンタクス

【図59】

Syntax	No. of bits	Mnemonics
AudioCodingInfo {		
audio_coding	8	uimsbf
audio_component_type	8	uimsbf
sampling_frequency	8	uimsbf
reserved	8	bsbf
}		

AudioCodingInfoのシンタクス

【図60】

audio_coding	Meaning
0	MPEG-1 audio layer I or II
1	Dolby AC-3 audio
2	MPEG-2 AAC
3	MPEG-2 multi-channel audio, backward compatible to MPEG-1
4	SESF LPCM audio
5-254	reserved
255	No information

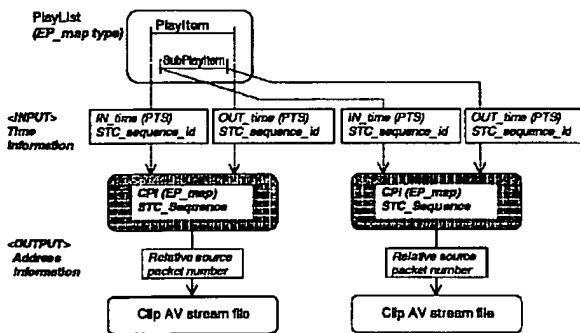
audio_coding

【図76】

Mark type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlaylistMark[]
0x90	Event-start mark	番組の開始ポイントを示すマーク点。
0x91	Local event-start mark	番組の中の局所的な場面を示すマーク点。
0x92	Scene-start mark	シーンチェンジポイントを示すマーク。
0x93 - 0xFF	reserved	

mark_type

【図63】

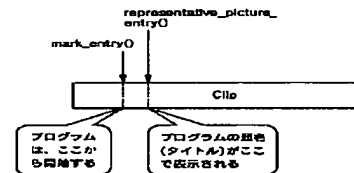


【図65】

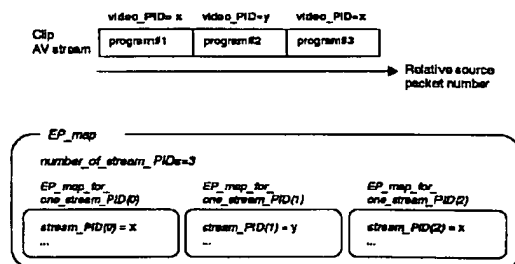
Syntax	No. of bits	Mnemonics
CPI {		
version number	8*4	bsbf
length	32	uimsbf
reserved	15	bsbf
CPI_type	1	bsbf
if (CPI_type == 0)		
EP_map()		
else		
TU_map()		
}		

CPIのシンタクス

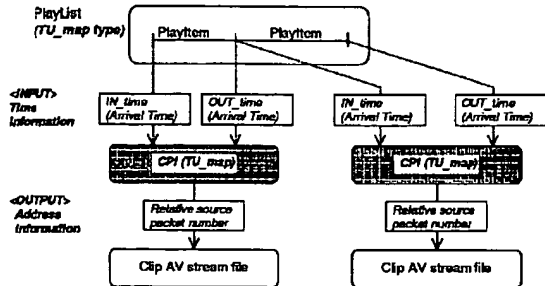
【図80】



【図69】



【図64】

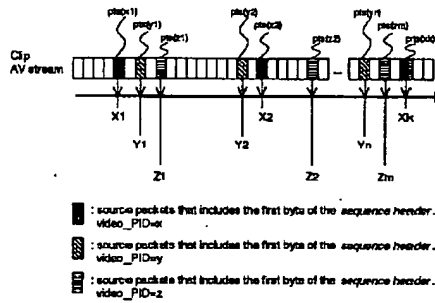


【図88】

Thumbnail picture format	Meaning
0x00	MPEG-2 Video I-picture
0x01	DCF (restricted JPEG)
0x02	PNG
0x03-0xff	reserved

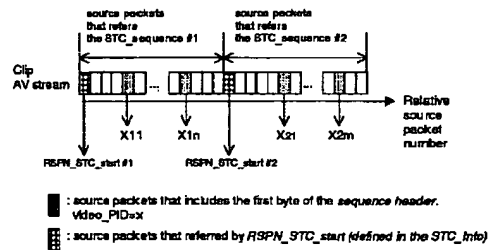
thumbnail_picture_format

【図67】

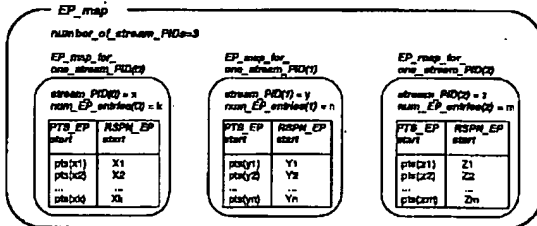


- : source packets that includes the first byte of the sequence header. video_PID=x
- : source packets that includes the first byte of the sequence header. video_PID=y
- : source packets that includes the first byte of the sequence header. video_PID=z

【図68】



- : source packets that includes the first byte of the sequence header. video_PID=x
- : source packets that referred by RSPN_STC_start (defined in the STC_Info)



ビデオの EP_map の例

EP_map for one stream PID

PTS_EP_start	RSPN_EP_start
pseq(1)	X11
...	...
pseq(1n)	X1n
...	...
pseq(21)	X21
...	...
pseq(2m)	X2m

RSPN_STC_start #2 < X21

【図78】

【図72】

Syntax	No. bits	of	Mnemonics
EP_map for one stream PID(N){			
for (i=0; i< N; i++){			
PTS_EP_start	32		uimsof
RSPN_EP_start	32		uimsof
}			

Syntax	No. bits	of	Mnemonics
ClipMarkQ {			
version_number	8*4		bsbf
length	32		uimsof
number of Clip marks	16		uimsof
for (i=0; i< number of Clip marks; i++){			
reserved	8		bsbf
mark_type	8		bsbf
reserved for MarkID	16		bsbf
mark_entry()			
representative_picture_entry()			
ref_thumbnail_index	16		uimsof
}			

EP_map_for_one_stream_PID のシンタクス

【 表 7 0 】

Syntax	No. of bits	Mnemonics
EP_map() {		
reserved	12	bsbf
EP_type	4	ulmsbf
number_of_stream_PIDs	16	ulmsbf
for (k=0; k<number_of_stream_PIDs; k++) {		
stream_PID(k)	16	bsbf
num_EP_entries(k)	32	ulmsbf
EP_map_for_one_stream_PID_Start_address(k)	32	ulmsbf
}		
for (i=0; i<X; i++) {		
padding_word	16	bsbf
}		
for (k=0; k<number_of_stream_PIDs; k++) {		
EP_map_for_one_stream_PID(num_EP_entries(k))		
for (j=0; j<Y; j++) {		
padding_word	16	bsbf
}		
}		
}		

【 表 7 1 】

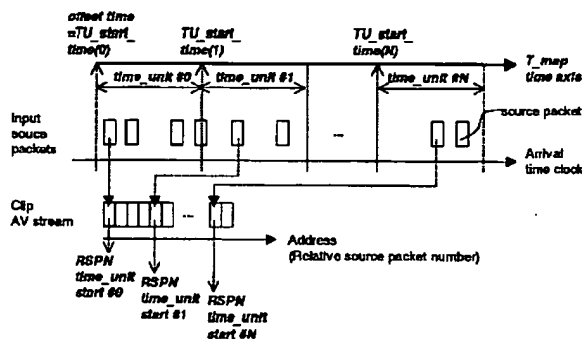
EP_type	Meaning
0	video
1	audio
2-15	reserved

EP_type Values

【 表 8 1 】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry() {		
mark_time_stamp	32	ulmsbf
STC_sequence_id	8	ulmsbf
reserved	24	bsbf
}		

【 表 7 3 】



【 表 7 9 】

Mark_type	Meaning	Comments
0x00 - 0x8F	reserved	Reserved for PlaylistMark()
0x90	Event-start mark	番組の開始ポイントを示すマーク点
0x91	Local event-start mark	番組の中での局所的な場面を示すマーク点
0x92	Scene-start mark	シーン開始ポイントを示すマーク点
0x93	Scene-end mark	シーン終了ポイントを示すマーク点
0x94	CM-start mark	CM 開始ポイントを示すマーク点
0x95	CM-end mark	CM 終了ポイントを示すマーク点
0x96 - 0xBF	DVR フォーマットが、ClipMark を得た、拡張するのために予約されている	
0xC0 - 0xFF	メーカー独自のアプリケーションで利用するマークに割り当て可能	

【 表 7 4 】

Syntax	No. of bits	Mnemonics
TU_map() {		
offset_time	32	bsbf
time_unit_size	32	ulmsbf
number_of_time_unit_entries	32	ulmsbf
for (k=0; k<number_of_time_unit_entries; k++)		
RSPN_time_unit_start	32	ulmsbf
}		

TU_map のシンタックス

【 表 7 5 】

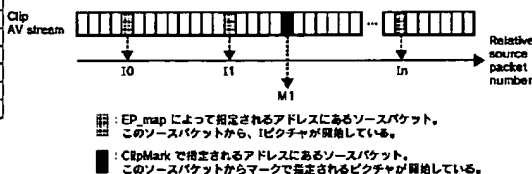
Syntax	No. of bits	Mnemonics
ClipMark() {		
version number	8*4	bslbf
length	32	uimsbf
number of Clip marks	16	uimsbf
for(i=0; i < number of Clip marks; i++) {		
reserved	8	bslbf
mark type	8	bslbf
mark time stamp	32	uimsbf
STC sequence id	8	uimsbf
reserved	24	bslbf
character set	8	bslbf
name length	8	uimsbf
mark name	8*256	bslbf
ref thumbnail index	16	uimsbf
}		
}		

ClipMark のシンタックス

【 表 8 2 】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry() {		
RSPN_ref_EP_start	32	uimsbf
offset_num_pictures	32	uimsbf
}		

【 表 8 5 】

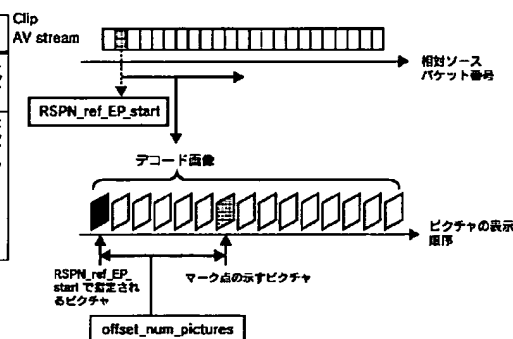


【 表 7 7 】

CPI_type in the CFI	Semantics of mark_time_stamp
EP_map type	mark_time_stamp は、マークで参照されるプレゼンテーションユニットに対応する 33 ビット長の PTS の上位 32 ビットを示さなければならない。
TU_map type	mark_time_stamp は、TU_map_time_axis 上の時刻でなければならない。かつ、mark_time_stamp は、time_unit の精度に丸めて表さなければならない。mark_time_stamp は、次に示す等式により計算される。 $\text{mark_time_stamp} = \text{TU_start_time} \% 2^{32}$

mark_type_stamp

【 表 8 3 】



【 表 8 4 】

Syntax	No. of bits	Mnemonics
mark_entry() / representative_picture_entry() {		
RSPN_mark_point	32	uimsbf
}		

【 表 8 6 】

【 表 9 5 】

copy_permission Indicator	meaning
00	copy free
01	no more copy
10	copy once
11	copy prohibited

copy permission indicator table

Syntax	No. of bits	Mnemonics
menu_thmb / mark_thmb {		
reserved	256	bslbf
Thumbnail() {		
for(i=0; i < Nt; i++)		
padding_word	16	bslbf
}		
}		

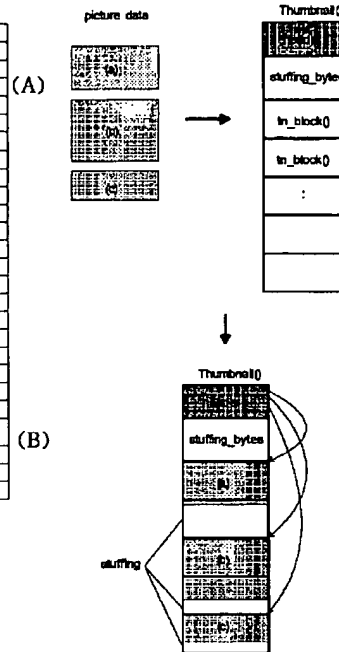
menu_thmb と mark_thmb のシンタックス

【 図 87 】

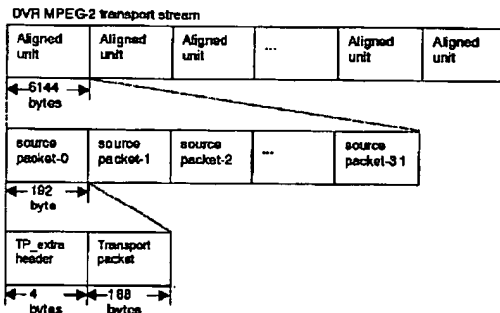
Syntax	Bits	Mnemonics
Thumbnail() {		
version_number	8*4	char
length	32	uimsbf
if (length != 0) {		
tn_blocks_start_address	32	bslbf
number_of_thumbnails	16	uimsbf
tn_block_size	16	uimsbf
number_of_tn_blocks	16	uimsbf
reserved	16	bslbf
for(i = 0; i < number_of_thumbnails; i++) {		
thumbnail_index	16	uimsbf
thumbnail_picture_format	8	bslbf
reserved	8	bslbf
picture_data_size	32	uimsbf
start_tn_block_number	16	uimsbf
x_picture_length	16	uimsbf
y_picture_length	16	uimsbf
reserved	16	uimsbf
}		
stuffing_bytes	0*2*L1	bslbf
for(k = 0; k < number_of_tn_blocks; k++) {		
tn_block	tn_block_size*1024*8	
}		
}		

Thumbnail の シンタクス

【 図 89 】

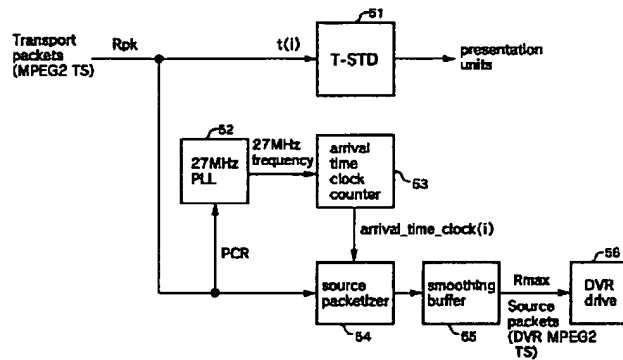


【 図 90 】



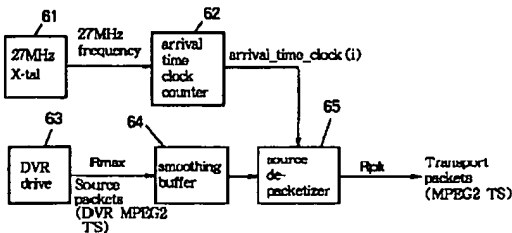
DVR MPEG-2 トランスポートストリームの構造

【 図 91 】



DVR MPEG-2 トランスポートストリームのレコーダモデル

【 図 93 】



DVR MPEG-2 トランスポートストリームのプレーヤモデル

Syntax	No. of bits	Mnemonics
source_packet() {		
TP_extra_header()		
transport_packet()		
}		

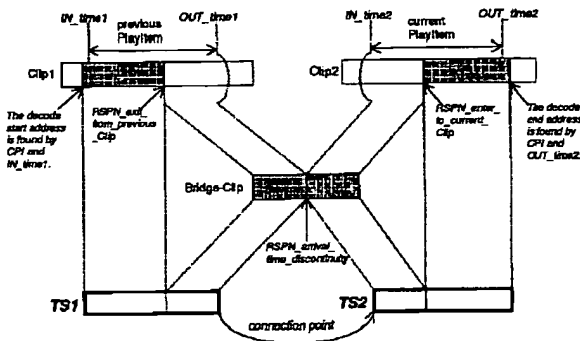
source packet

【図94】

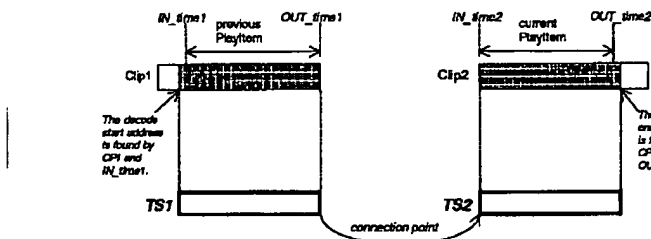
Syntax	No. bits	cf	Mnemonics
TP extra header{			
copy permission indicator	2		uimbsf
arrival time stamp	30		uimbsf
}			

TP_extra_header

【図96】



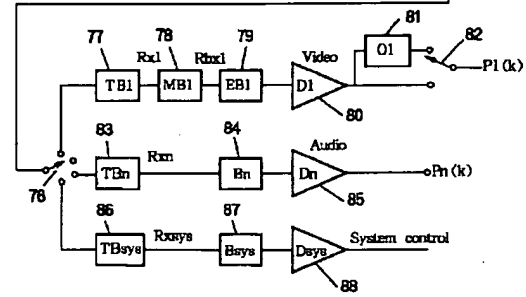
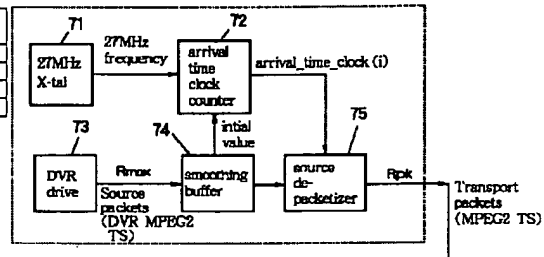
【図97】



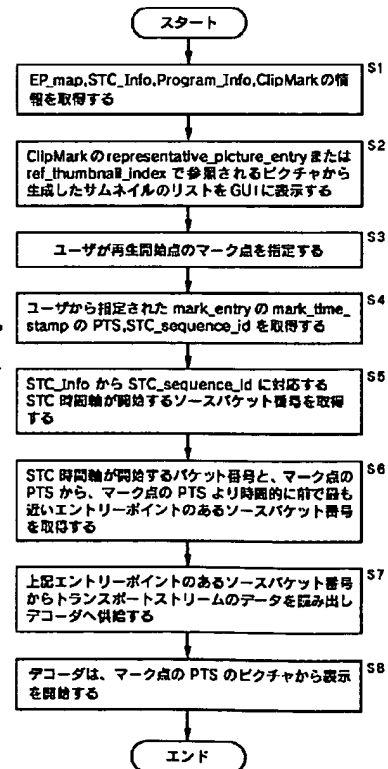
【図109】

ClipMark	mark_entry		representative_picture_entry	
	Mark_type	Mark_time_stamp	Mark_time_stamp	STC_sequence_id
0x92(scene start)	PTS(a1)	id0	PTS(a2)	id0
0x94(CMstart)	PTS(b0)	id0	PTS(b0)	id0
0x95(CMend)	PTS(c0)	id0	PTS(c0)	id0
...

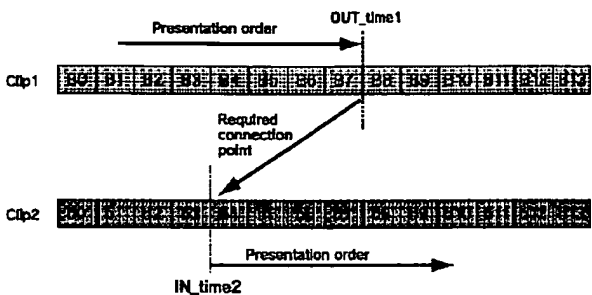
【図104】



【图 106】

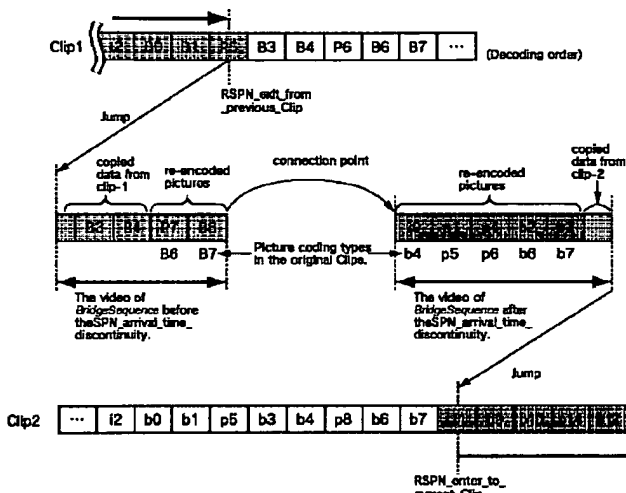


【図98】



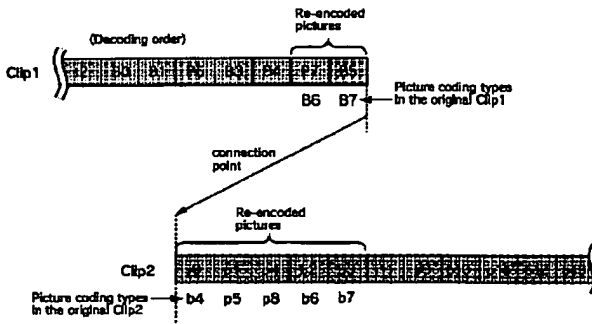
ピクチャの表示順序で示すシームレス接続の例

【図99】



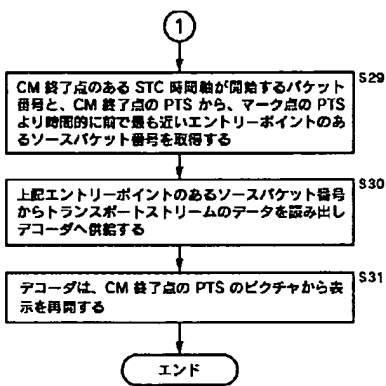
BridgeSequence を使用してシームレス接続を実現する例1

【図100】

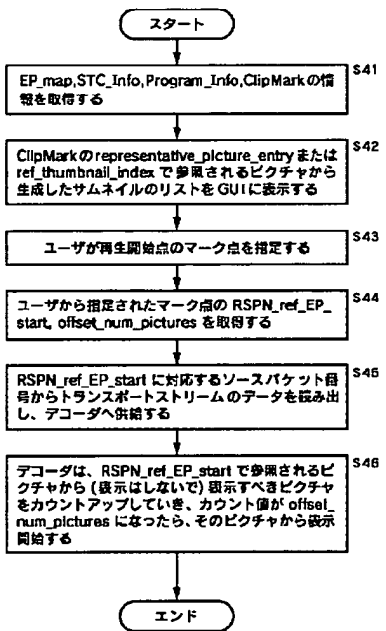


BridgeSequence を使用しないでシームレス接続を実現する例2

【図111】



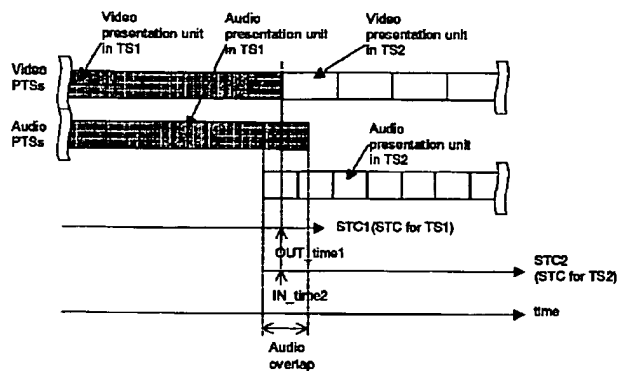
【図112】



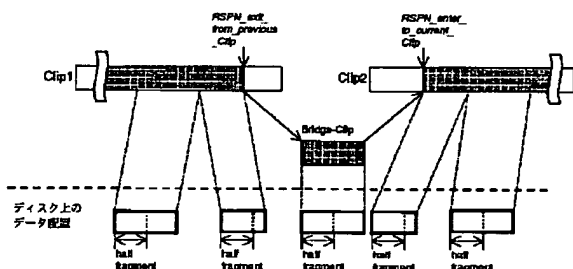
【図115】

mark_type	mark_entry		representative_picture_entry	
	RSPN_ref_EP_start	offset_num_pictures	RSPN_ref_EP_start	offset_num_pictures
...
0x92(scene start)	A	M1	A	M2
0x94(CM start)	B	N1	B	N1
0x95(CM end)	C	N2	C	N2
...

【図101】

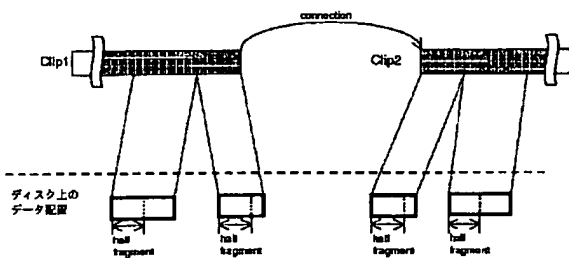


【図102】



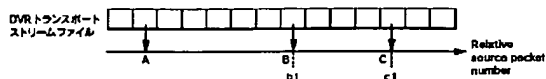
BridgeSequence を使用してシームレス接続をする場合の、データアロケーションの例

【図103】

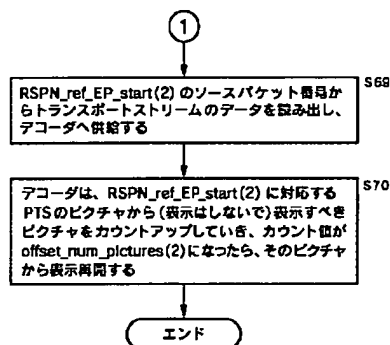


BridgeSequence を使用しないでシームレス接続をする場合の、データアロケーションの例

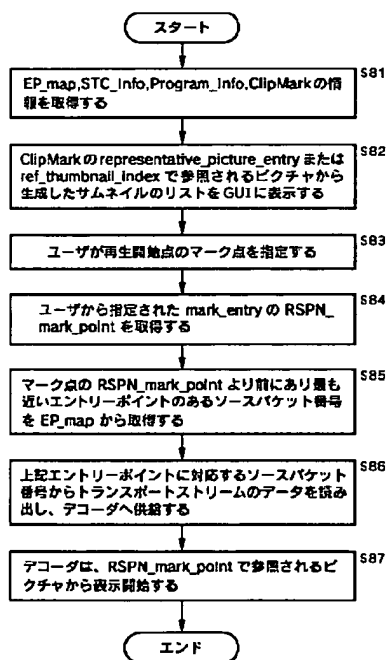
【図119】



【図117】



【図118】

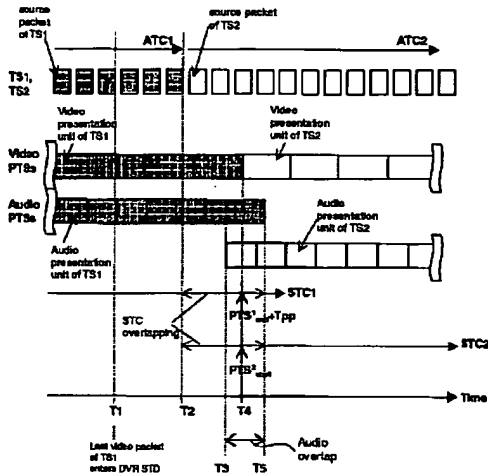


【図121】

ClipMark

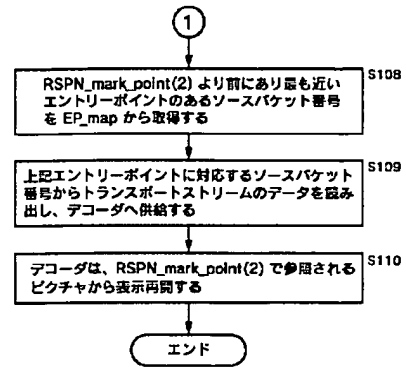
mark_type	mark_entry RSPN_mark_point	representative_picture_entry RSPN_mark_point
...
0x92(scene start)	a 1	a 2
0x94(CM start)	b 1	b 1
0x96(CM end)	c 1	c 1
...

【図105】

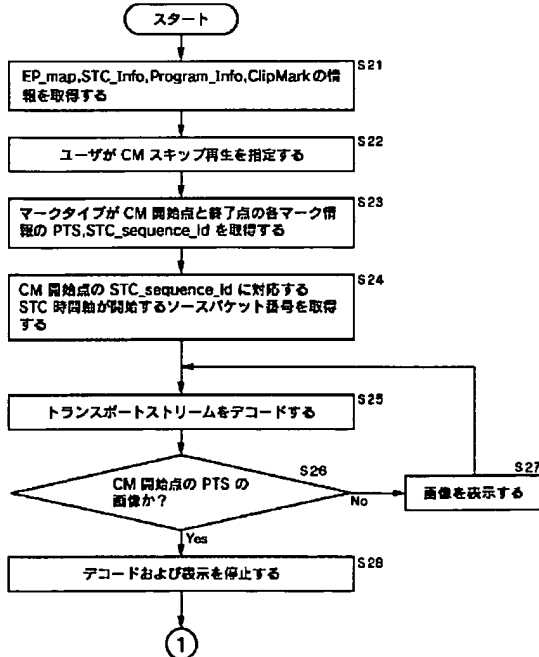


ある AV ストリーム(TS1)からそれにシームレスに接続された次の AV ストリーム(TS2)へと移る時のトランスポートパケットの入力、復号、表示のタイミングチャート

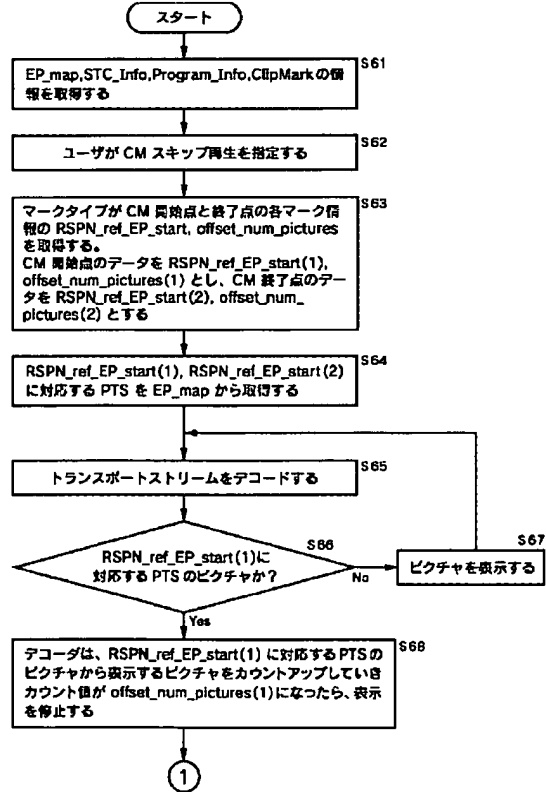
【図123】



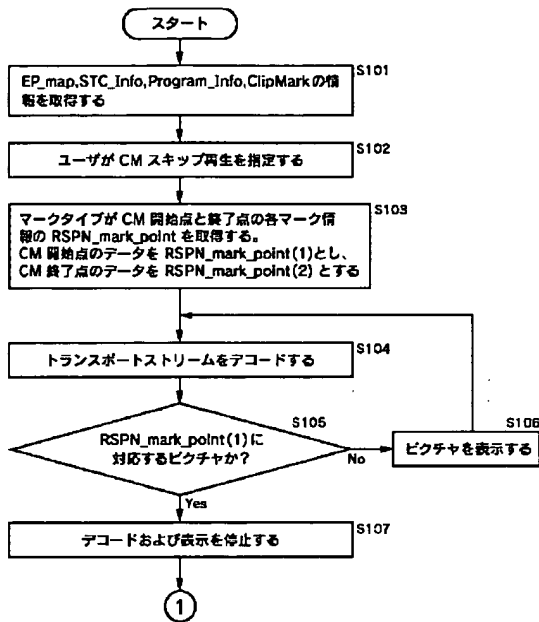
【図110】



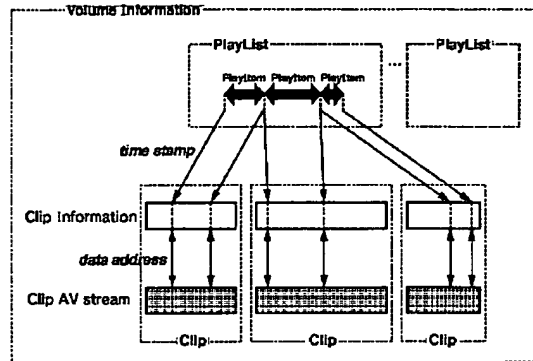
【図116】



【 図 122 】

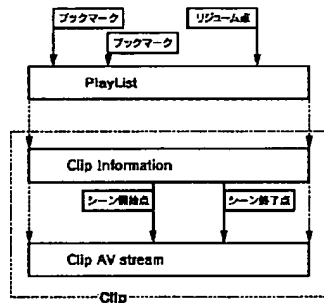


【 図 124 】



【 図 125 】

PlayListMark
タイムスタンプベースの指定
(例) リジューム点、ブックマーク



ClipMark
アドレスベースの指定
(例) シーンスタート点、
シーン終了点

【 図 126 】

Syntax	No. of bits	Mnemonics
ClipMark0 {		
version_number	8*4	bsbtf
length	32	uimbsf
number_of_Clip_marks	16	uimbsf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bsbtf
mark_type	8	bsbtf
RSPN_mark	32	uimbsf
reserved	32	bsbtf
ref_thumbnail_index	16	uimbsf
}		
}		

【 図 128 】

【 図 127 】

Syntax	No. of bits	Mnemonics
ClipMark0 {		
version_number	8*4	bsbtf
length	32	uimbsf
number_of_Clip_marks	16	uimbsf
for (i=0; i<number_of_Clip_marks; i++){		
reserved	8	bsbtf
mark_type	8	bsbtf
RSPN_ref_EP_start	32	uimbsf
offset_num_pictures	32	uimbsf
ref_thumbnail_index	16	uimbsf
}		
}		

Syntax	No. of bits	Mnemonic
ClipInfo0 {		
Length	32	uimbsf
reserved_for_word_align	8	bsbtf
Clip_service_type	8	uimbsf
Clip_stream_type	8	uimbsf
reserved_for_word_align	6	bsbtf
transcode_mode_flag	1	bsbtf
time_controlled_flag	1	bsbtf
TS_average_rate	32	uimbsf
TS_recording_rate	32	uimbsf
reserved_for_DVRsystem_use	144	bsbtf
TS_type_Info_block()		
}		

ClipInfo0 のシンタックスの例

【図129】

Syntax	No. of bits	Mnemonic
ProgramInfo() {		
length	32	uimsbf
reserved_for_word_align	8	bslbf
num_of_program_sequences	8	uimsbf
for (i=0; i<num_of_program_sequences; i++){		
SPN_program_sequences_start	32	uimsbf
program_map_PID	16	bslbf
num_of_streams_in_ps	8	uimsbf
num_of_groups	8	uimsbf
for (stream_index=0; stream_index<num_of_streams_in_ps; stream_index++){		
stream_PID	16	uimsbf
StreamCodingInfo()		
}		
if (num_of_groups>1){		
for (i=0; i<num_of_groups; i++){		
num_of_streams_in_group	8	uimsbf
for (k=0; k<num_of_streams_in_group; k++){		
stream_index	8	uimsbf
}		
if (num_of_streams_in_group%2==0){		
reserved_for_word_align	8	bslbf
}		
}		
}		
}		

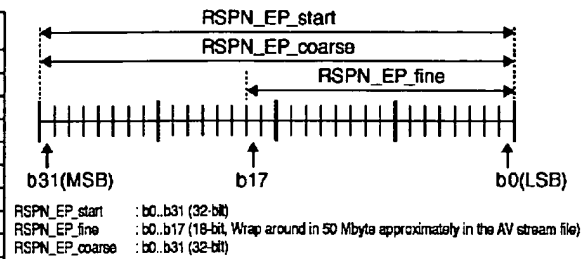
ProgramInfo() のシンタクスの例

【図130】

Syntax	No. of bits	Mnemonic
StreamCodingInfo() {		
length	8	bslbf
stream_coding_type	8	uimsbf
if (stream_coding_type==0x02){		
video_format	4	uimsbf
frame_rate	4	uimsbf
display_aspect_ratio	4	uimsbf
reserved_for_word_align	2	bslbf
cc_flag	1	uimsbf
original_video_format_flag	1	uimsbf
if (original_video_format_flag==1){		
original_video_format	4	uimsbf
original_display_aspect_ratio	4	uimsbf
reserved_for_word_align	8	bslbf
}		
} else if (stream_coding_type==0x03 // stream_coding_type==0x04 // stream_coding_type==0x0F // stream_coding_type==0x80 // stream_coding_type==0x81 //		
audio_presentation_type	4	uimsbf
sampling_frequency	4	uimsbf
reserved_for_word_align	8	bslbf
}		
}		

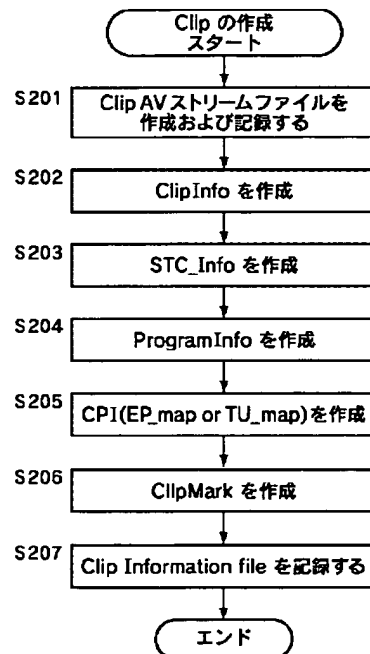
StreamCodingInfo() のシンタクス

【図134】



RSPN_EP_fine と RSPN_EP_coarse のフォーマットを説明する図

【図140】

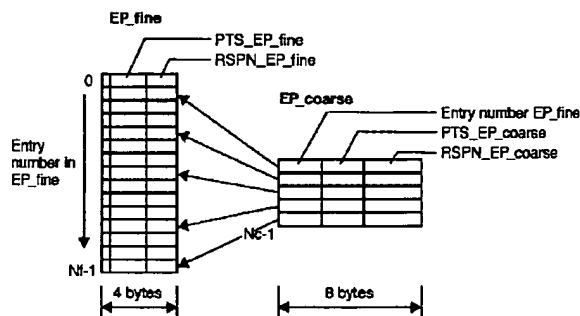


Clip AV ストリームファイルおよび Clip Information ファイルの作成

【図131】

stream_coding_type	Meaning
0x00 - 0x01	reserved for future use
0x02	MPEG-1 or MPEG-2 video stream
0x03	MPEG-1 audio
0x04	MPEG-2 multi-channel audio, backward compatible to MPEG-1
0x05	reserved for future use
0x06	Teletext defined in SESF or DVB or Subtitle defined in ISDB
0x07 - 0x09	reserved for future use
0x0A	ISO/IEC 13818-6 type A
0x0B	ISO/IEC 13818-6 type B
0x0C	ISO/IEC 13818-6 type C
0x0D	ISO/IEC 13818-6 type D
0x0E	reserved for future use
0x0F	MPEG-2 AAC audio with ADTS transport syntax
0x10 - 0x7F	reserved for future use
0x80	SESF LPCM audio
0x81	Dolby AC-3 audio
0x82 - 0xFF	reserved for future use

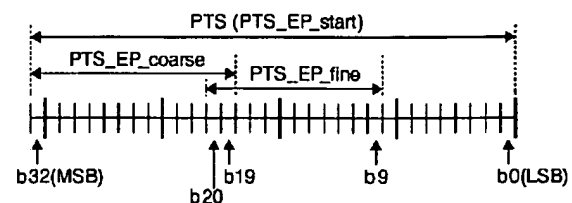
【図132】



Nf is the number of entries in EP_fine
Nc is the number of entries in EP_coarse (Nc < Nf)

EP-fine と EP-coarse の関係を説明する図

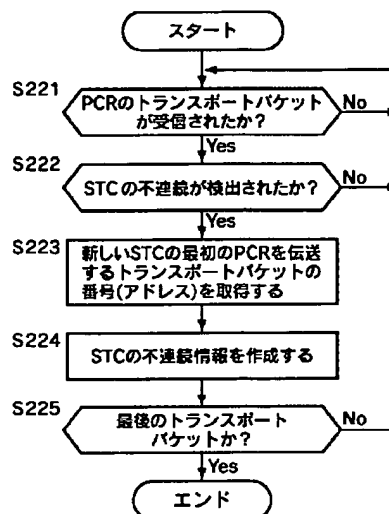
【図133】



PTS : b0..b32(33-bit, 90kHz)
PTS_EP_fine : b9..b20(12-bit, Resolution=5.7msec and Wraparound in 23 seconds approximately)
PTS_EP_coarse : b19..b32(14-bit, Resolution=5.8sec and Wraparound in 26.5 hours approximately)

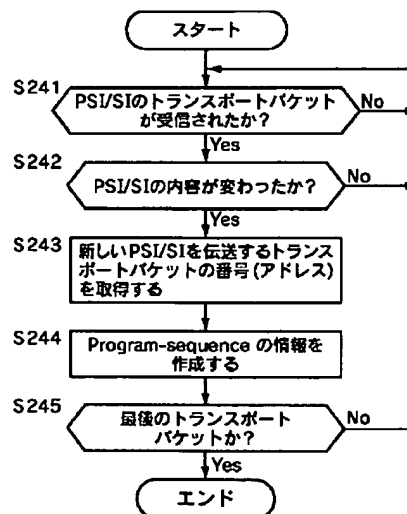
PTS_EP_fine と PTS_EP_coarse のフォーマットを説明する図

【図141】



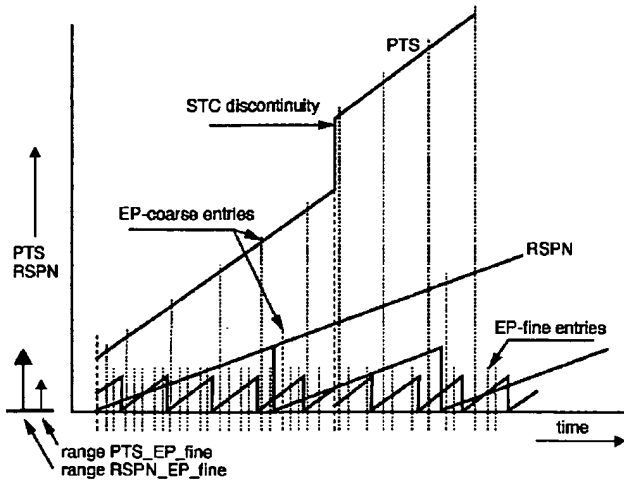
STC_Info の作成の動作例を説明するフローチャート

【図142】



Program_Info の作成の動作例を説明するフローチャート

【図135】



EP-coarse のエントリーと EP-fine のエントリーを説明する図

【図136】

Syntax	No. of bits	Mnemonic
EP_map() {		
reserved_for_word_align	8	bslbf
number_of_stream_PID_entries	8	uimsbf
for (k=0; k<number_of_stream_PID_entries; k++){		
stream_PID[k]	16	bslbf
reserved_for_word_align	10	bslbf
EP_stream_type[k]	4	uimsbf
num_EP_coarse_entries[k]	16	uimsbf
num_EP_fine_entries[k]	18	uimsbf
EP_map_for_one_stream_PID_start_address[k]	32	uimsbf
}		
for (l=0; l<X; l++){		
padding_word	16	bslbf
}		
for (k=0; k<number_of_stream_PID_entries; k++){		
EP_map_for_one_stream_PID (EP_stream_type[k],		
num_EP_coarse_entries[k],		
num_EP_fine_entries[k])		
for (l=0; l<Y[k]; l++){		
padding_word	16	bslbf
}		
}		
}		

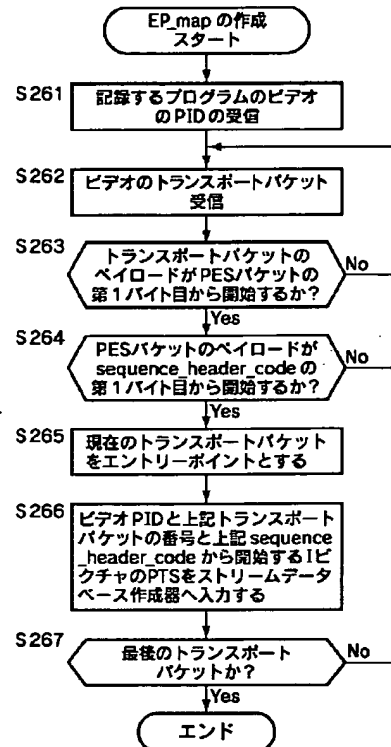
EP_map のシンタックスの例

【図137】

EP_stream_type values

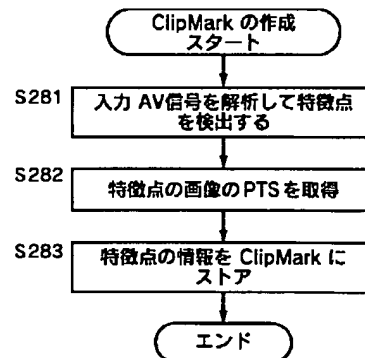
EP_stream_type	Meaning
0	video type1
1	video type2
2	audio
3-15	reserved for future use

【図143】



EP_map の作成の動作例を説明するフローチャート

【図144】



アナログAV信号をエンコードして記録する場合における、図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図81のシンタックスであるときのClipMarkの作成方法を説明するフローチャート

【図138】

Syntax	No. of bits	Mnemonic
EP_map_for_one_stream_PID(EP_stream_type,Nc,Nf){		
EP_fine_table_start_address	32	uimsbf
for (i=0; i<Nc; i++){		
ref_to_EP_fine_id[i]	18	uimsbf
PTS_EP_coarse[i]	14	uimsbf
RSPN_EP_coarse[i]	32	uimsbf
}		
for (i=0; i<X; i++){		
padding_word	16	bslbf
}		
for (EP_fine_id = 0;		
EP_fine_id < Nf;		
EP_fine_id ++){		
EP_video_type[EP_fine_id]	2	
PTS_EP_fine[EP_fine_id]	12	uimsbf
RSPN_EP_fine[EP_fine_id]	18	uimsbf
}		
}		

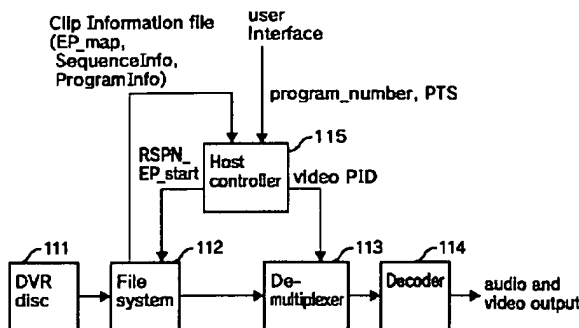
図136のEP_mapのEP_map_for_one_stream_PIDのシンタクス

【図139】

value	Meaning
0	エントリーポイントにあたるビデオアクセスユニットは、シーケンスヘッダから始まるIピクチャである、またそのIピクチャの前にGOPヘッダがあっても良い。SPN_EP_startは、そのアクセスユニットのシーケンスヘッダコードの第1バイト目を含むソースパケットのアドレスを示す。
1	エントリーポイントにあたるビデオアクセスユニットは、シーケンスヘッダから始まるPピクチャである。SPN_EP_startは、そのアクセスユニットのシーケンスヘッダコードの第1バイト目を含むソースパケットのアドレスを示す。
2	エントリーポイントにあたるビデオアクセスユニットは、シーケンスヘッダから開始しないIピクチャである、またそのIピクチャの前にGOPヘッダがあっても良い。Iピクチャの前にGOPヘッダがある場合、SPN_EP_startは、そのアクセスユニットのグループ・スタート・コードの第1バイト目を含むソースパケットのアドレスを示す。Iピクチャの前にGOPヘッダがない場合、SPN_EP_startは、そのアクセスユニットのグループ・スタート・コードの第1バイト目を含むソースパケットのアドレスを示す。
3	reserved for future use

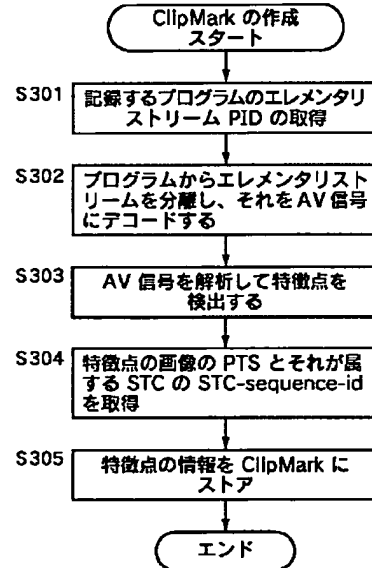
EP_video_type の値の意味

【図147】



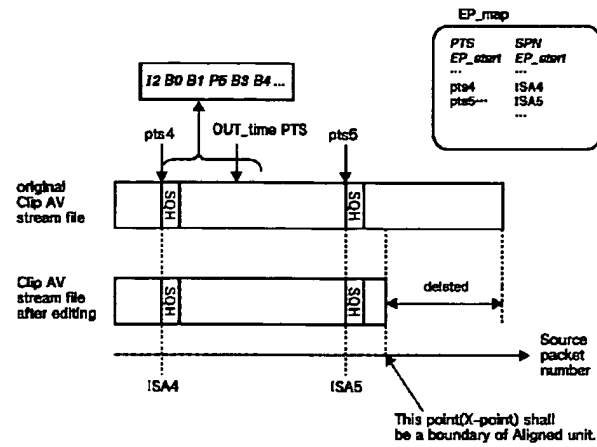
EP_map を使用した Iピクチャサーチのためのプレーヤモデル

【図145】



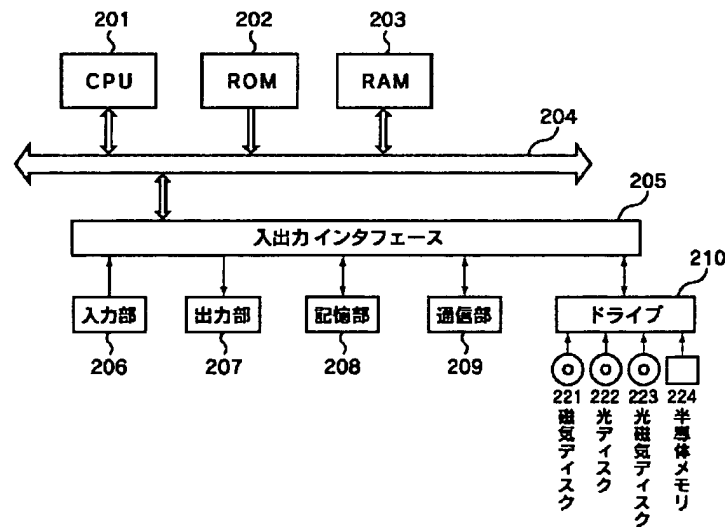
デジタルインタフェースから入力されたトランスポートストリームを記録する場合における、図75または図78のClipMarkのmark_entry()/representative_picture_entry()が図81のシンタクスであるときのClipMarkの作成方法を説明するフローチャート

【図150】



ミニマイズの時に OUT_time の後ろの不要なストリームデータを消去する例

【図151】



フロントページの続き

Fターム(参考) 5C053 FA14 FA23 GB06 GB08 GB15
 GB38 HA21 HA24 JA21 JA24
 KA26 LA01 LA11
 5D044 AB05 AB07 BC04 CC06 DE12
 DE38 DE39 DE49 DE53 EF05
 FG18 GK08
 5D110 AA17 AA19 AA26 AA28 DA02
 DA11 DA15 DA17 DA19 DB03
 DB09 DC05 DC16